

TRABAJO FIN DE GRADO



Virtual Ants

Hormigas virtuales y paradigmas de búsqueda sociales



UNIVERSIDAD CARLOS III DE MADRID
Grado en Ingeniería Informática
Curso 2012/2013

Tutor: Alessio Malizia
Autor: Christian Cabañas Panes

Junio 2013

ÍNDICE

1. INTRODUCCIÓN	8
1.1. Planteamiento del problema	8
1.2. Objetivos	9
1.3. Estructura del documento	10
2. ESTADO DEL ARTE	11
2.1. Algoritmo PageRank de Google	11
2.1.1. <i>Alternativas a PageRank</i>	14
2.1.2. <i>Valoración de los algoritmos existentes</i>	15
2.2. Search Engine Optimization (SEO)	16
2.3. Algoritmos sociales	17
2.3.1. <i>Collaborative Filtering</i>	18
2.3.2. <i>EdgeRank</i>	18
2.3.3. <i>Particle Swarm Optimization (PSO)</i>	20
2.4. Ant Colony Optimization (ACO)	21
2.4.1. <i>Hormigas naturales</i>	21
2.4.1. <i>Hormigas virtuales</i>	23
2.5. Comparación PageRank - ACO	26
3. ANÁLISIS DE LA PROPUESTA	27
3.1. Propuesta	27
3.1.1. <i>Herramienta propuesta basada en ACO</i>	27
3.1.2. <i>Plataforma propuesta</i>	31
3.2. Condiciones de entorno	32
3.3. Especificación de requisitos	32
3.3.1. <i>Requisitos de usuario</i>	32
3.3.2. <i>Casos de uso</i>	36
3.3.3. <i>Requisitos de software</i>	38
3.3.4. <i>Matriz de trazabilidad</i>	48

4. DISEÑO E IMPLEMENTACIÓN	50
4.1. Diseño.....	50
4.1.1. <i>Arquitectura del sistema</i>	50
4.1.2. <i>Tecnologías empleadas</i>	51
4.1.3. <i>Entorno de desarrollo</i>	55
4.1.4. <i>Entorno de implantación</i>	57
4.2. Implementación.....	58
4.2.1. <i>Extensión de Chrome</i>	58
4.2.2. <i>Lógica de negocio</i>	64
4.2.3. <i>Base de datos</i>	65
5. MARCO REGULADOR	67
6. VALIDACIÓN DEL SISTEMA	69
6.1. Pruebas funcionales	70
6.2. Pruebas de comunicación.....	75
6.3. Pruebas de aceptación	76
7. GESTIÓN DEL PROYECTO	77
7.1. Planificación	77
7.1.1. <i>Creación del Plan de Proyecto</i>	78
7.1.2. <i>Diagrama de Gantt</i>	78
7.2. Presupuesto	83
7.2.1. <i>Recursos materiales</i>	83
7.2.2. <i>Recursos humanos</i>	83
7.2.3. <i>Viajes y dietas</i>	84
7.2.4. <i>Gastos indirectos</i>	85
7.2.5. <i>Presupuesto total</i>	85
8. CONCLUSIONES Y TRABAJO FUTURO	86
8.1. Conclusiones.....	86
8.2. Trabajo futuro	87

9. BIBLIOGRAFÍA	88
ANEXO I: ACRÓNIMOS Y ABREVIATURAS	91
ANEXO II: MANUAL DE USUARIO	92
Instalación.....	92
Empezar a utilizar la extensión	93
Realizar una búsqueda en Google.....	95
Acciones sobre las páginas	95
ANEXO III: POLÍTICA DE PRIVACIDAD	96

ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1. Top 5 en Motores de Búsqueda (StatCounter 2012)</i>	11
<i>Ilustración 2. Ejemplo gráfico PageRank</i>	13
<i>Ilustración 3. Comportamiento de las hormigas en busca de una ruta al alimento</i>	22
<i>Ilustración 4. Nomenclatura requisitos de usuario</i>	32
<i>Ilustración 5. Diagrama de casos de uso</i>	36
<i>Ilustración 6. Nomenclatura requisitos de software</i>	38
<i>Ilustración 7. Arquitectura del sistema</i>	50
<i>Ilustración 8. Estructura general de un documento HTML</i>	52
<i>Ilustración 9. Página de administración de extensiones (Modo desarrollador)</i>	56
<i>Ilustración 10. Ventana para Empaquetar extensión</i>	57
<i>Ilustración 11. Icono de Virtual Ants</i>	59
<i>Ilustración 12. Página de Inicio (Virtual Ants)</i>	60
<i>Ilustración 13. Página de Bienvenida (Virtual Ants)</i>	60
<i>Ilustración 14. Autenticación incorrecta (Virtual Ants)</i>	61
<i>Ilustración 15. Página de Registro (Virtual Ants)</i>	61
<i>Ilustración 16. Página de Resultados de Google (Virtual Ants)</i>	62
<i>Ilustración 17. Ciclo de vida del Trabajo Fin de Grado</i>	77
<i>Ilustración 18. Diagrama de Gantt (1)</i>	80
<i>Ilustración 19. Diagrama de Gantt (2)</i>	81
<i>Ilustración 20. Diagrama de Gantt (3)</i>	82

ÍNDICE DE ECUACIONES

<i>Ecuación 1. Algoritmo PageRank inicial (1998)</i>	12
<i>Ecuación 2. Algoritmo HITS</i>	15
<i>Ecuación 3. Algoritmo EdgeRank</i>	19
<i>Ecuación 4. Regla probabilística de transición (ACO)</i>	24
<i>Ecuación 5. Actualización de la feromona (ACO)</i>	25
<i>Ecuación 6. Aporte de feromona (Virtual Ants)</i>	30
<i>Ecuación 7. Evaporación de feromona (Virtual Ants)</i>	30

ÍNDICE DE TABLAS

Tabla 1. Comparación PageRank - ACO	26
Tabla 2. UC-001	33
Tabla 3. UC-002	33
Tabla 4. UC-003	34
Tabla 5. UC-004	34
Tabla 6. UC-005	34
Tabla 7. UR-001	34
Tabla 8. UR-002	35
Tabla 9. UR-003	35
Tabla 10. UR-004	35
Tabla 11. UR-005	35
Tabla 12. Caso de uso - Agregar Extensión	37
Tabla 13. Caso de uso - Registrarse	37
Tabla 14. Caso de uso - Iniciar Sesión	37
Tabla 15. Caso de uso - Buscar en Google	37
Tabla 16. Caso de uso - Ver resultados alternativos	37
Tabla 17. Caso de uso - Dejar feromonas	38
Tabla 18. SF-001	39
Tabla 19. SF-002	39
Tabla 20. SF-003	40
Tabla 21. SF-004	40
Tabla 22. SF-005	40
Tabla 23. SF-006	40
Tabla 24. SF-007	41
Tabla 25. SF-008	41
Tabla 26. SF-009	41
Tabla 27. SF-010	41
Tabla 28. SF-011	42
Tabla 29. SI-001	42
Tabla 30. SI-002	42
Tabla 31. SI-003	43
Tabla 32. SI-004	43
Tabla 33. SI-005	43
Tabla 34. SI-006	43
Tabla 35. SO-001	44
Tabla 36. SO-002	44
Tabla 37. SO-003	44
Tabla 38. SO-004	45
Tabla 39. SO-005	45
Tabla 40. SR-001	45
Tabla 41. SR-002	45
Tabla 42. SC-001	46
Tabla 43. SC-002	46
Tabla 44. SC-003	46
Tabla 45. SA-001	47
Tabla 46. SA-002	47
Tabla 47. SD-001	47
Tabla 48. SS-001	48

<i>Tabla 49. Matriz de trazabilidad. Requisitos usuario / Requisitos software</i>	<i>49</i>
<i>Tabla 50. Formato de especificación de prueba</i>	<i>69</i>
<i>Tabla 51. PR-F001</i>	<i>70</i>
<i>Tabla 52. PR-F002</i>	<i>70</i>
<i>Tabla 53. PR-F003</i>	<i>71</i>
<i>Tabla 54. PR-F004</i>	<i>71</i>
<i>Tabla 55. PR-F005</i>	<i>72</i>
<i>Tabla 56. PR-F006</i>	<i>72</i>
<i>Tabla 57. PR-F007</i>	<i>73</i>
<i>Tabla 58. PR-F008</i>	<i>73</i>
<i>Tabla 59. PR-F009</i>	<i>74</i>
<i>Tabla 60. PR-F010</i>	<i>74</i>
<i>Tabla 61. PR-F011</i>	<i>74</i>
<i>Tabla 62. PR-C001</i>	<i>75</i>
<i>Tabla 63. PR-C002</i>	<i>75</i>
<i>Tabla 64. PR-A001</i>	<i>76</i>
<i>Tabla 65. PR-A002</i>	<i>76</i>
<i>Tabla 66. Tareas identificadas en la planificación del proyecto</i>	<i>78</i>
<i>Tabla 67. Resumen de costes en recursos materiales</i>	<i>83</i>
<i>Tabla 68. Resumen de costes en recursos humanos</i>	<i>84</i>
<i>Tabla 69. Resumen de costes de viajes y dietas</i>	<i>84</i>
<i>Tabla 70. Resumen de costes indirectos</i>	<i>85</i>
<i>Tabla 71. Resumen del Presupuesto Total</i>	<i>85</i>

1. INTRODUCCIÓN

Comenzaremos con un breve texto exponiendo un problema y los objetivos para su resolución en este Trabajo Fin de Grado, así como el ámbito y alcance del mismo.

Finalizaremos este primer capítulo mostrando la estructura que presentará el documento, con una breve descripción sobre los contenidos del mismo.

1.1. Planteamiento del problema

Desde que en 1996 la WWW (World Wide Web) dejó de ser una herramienta reservada a la ciencia y la enseñanza, su crecimiento fue imparable. Esta apertura supuso que se crearan páginas con todo tipo de contenidos, generando una cantidad de información enorme.

El aumento del volumen de la información provocó que encontrar información sobre la que se tenía interés se hiciera cada vez más complicado. La Web facilitó la indexación, que consiste en ordenar y etiquetar el contenido de manera adecuada formando un índice.

De esta forma nacieron los motores de búsqueda. Su comportamiento se basa en usar programas rastreadores encargados de recopilar la información e incorporarlas a sus índices para ofrecer los resultados a los usuarios.

Por otra parte, los motores de búsqueda necesitan algoritmos para ofrecer los mejores resultados a esos usuarios en busca de algún tipo de información o servicio de su interés. El problema surge cuando se deben elegir los criterios a utilizar para decidir si una página es más importante que otra, es decir, cuando se debe calcular su relevancia.

Actualmente la popularidad web es la medida utilizada por todos los buscadores para clasificar los resultados de una búsqueda. La popularidad de una página web es calculada en base a la cantidad de enlaces que recibe, es decir, una página tendrá una popularidad mayor cuanto mayor sea el número de vínculos existentes a ella en otras páginas.

Para el cálculo de la popularidad los buscadores también tienen en cuenta la calidad de los enlaces, pero no del contenido. La calidad de los enlaces viene reconocida si el enlace está fuertemente ligado a los contenidos de la web enlazada, así como si el enlace pertenece a una página con una buena popularidad en las búsquedas que interesan.

Esta forma de calcular la relevancia tiene la desventaja de estar basados en una visión estática de la web. Esto quiere decir que los sitios web importantes cada vez lo serán más y que un sitio nuevo, el cual podría ser interesante, obtendrá una baja clasificación en una búsqueda debido a su menor popularidad.

Cuando éste nuevo sitio web contiene información de calidad para mostrar al usuario que realiza la búsqueda, surge el problema de que únicamente podrá ser visitado por los usuarios más persistentes y tendrá pocas posibilidades de aumentar su posición en el ranking, debido a esa visión estática de la web.

1.2. Objetivos

Una vez se ha expuesto el problema existente y por el cual se ha decidido realizar este Trabajo Fin de Grado, exponemos los objetivos establecidos para su solución.

El surgimiento de las redes sociales, donde personas conectadas interactúan compartiendo información, nos puede ofrecer una idea para saber donde existe información de calidad y de interés para sus usuarios.

Nuestro objetivo será tener en cuenta esa actividad social o la actividad de los usuarios de la web en general para calcular la relevancia de un sitio web. Pretendemos que cuando un usuario realice una consulta en un motor de búsqueda se le ofrezcan unos resultados en los que se han tenido en cuenta si un sitio web ha resultado relevante para otros usuarios con el mismo interés.

Proyectamos crear un prototipo de aplicación con la cual podríamos facilitar enlaces que han sido de interés para usuarios que realizaron una misma búsqueda, haciendo que los resultados mostrados no sean estáticos debido a su número de enlaces sino que puedan variar en función del interés creado.

En resumen, queremos que nuestra herramienta calcule la relevancia de un sitio web de una manera social y dinámica.

1.3. Estructura del documento

A continuación detallamos la estructura que va a seguir el presente documento y se realiza una breve descripción sobre el contenido de cada uno de sus capítulos:

- Capítulo 2. Estado del Arte
Análisis de los algoritmos existentes en la actualidad para el posicionamiento en motores de búsqueda. Presentación algoritmos sociales como alternativa.
- Capítulo 3. Análisis de la Propuesta
Enunciación de la solución propuesta para la resolución del problema. Exposición de los requisitos de usuario y software de la herramienta que se pretende desarrollar.
- Capítulo 4. Diseño e Implementación
Análisis de la arquitectura del sistema. Definición de las tecnologías empleadas. Codificación de la herramienta.
- Capítulo 5. Marco Regulator
Análisis de las restricciones aplicables al sistema.
- Capítulo 6. Validación del Sistema
Plan de pruebas realizado para comprobar el correcto funcionamiento del sistema.
- Capítulo 7. Gestión del Proyecto
Diagramas de planificación e hitos que se han seguido para su desarrollo. Desglose del coste que ha supuesto el desarrollo de la herramienta.
- Capítulo 8. Conclusiones y Trabajo Futuro
Exposición de las conclusiones alcanzadas tras la realización del trabajo. Presentación de posibles líneas de código futuras.
- Capítulo 9. Bibliografía
Referencias bibliográficas ordenadas por orden de aparición en el documento.
- Anexo I. Acrónimos y Abreviaturas
Acrónimos utilizados a lo largo del documento, ordenados alfabéticamente.
- Anexo II. Manual de Usuario
Instrucciones de instalación y uso de la herramienta desarrollada.
- Anexo III. Política de Privacidad
Política que deberán aceptar los usuarios para el registro en la herramienta.

2. ESTADO DEL ARTE

En este apartado se presentan las investigaciones realizadas sobre el tema que nos trata, SEO (Search Engine Optimization) que significa *posicionamiento en los motores de búsqueda*.

Expondremos el algoritmo utilizado por Google para calcular el posicionamiento de una página y las modificaciones que en los últimos años ha sufrido el algoritmo inicial para mejorar los resultados ofrecidos a sus usuarios.

Seguidamente realizaremos un análisis de las alternativas a PageRank existentes en el mercado y su modo de calcular la relevancia de un sitio Web.

Y para finalizar mostraremos las investigaciones que se están llevando a cabo basadas en algoritmos de carácter social.

2.1. Algoritmo PageRank de Google

Entre todos los motores de búsqueda existentes en el mercado, Google es en la actualidad el más utilizado por la mayoría de los usuarios de Internet, según se muestra claramente en la siguiente gráfica, ofrecida por StatCounter¹, que identifica la tendencia de uso en los motores de búsqueda en el año 2012 a nivel mundial.

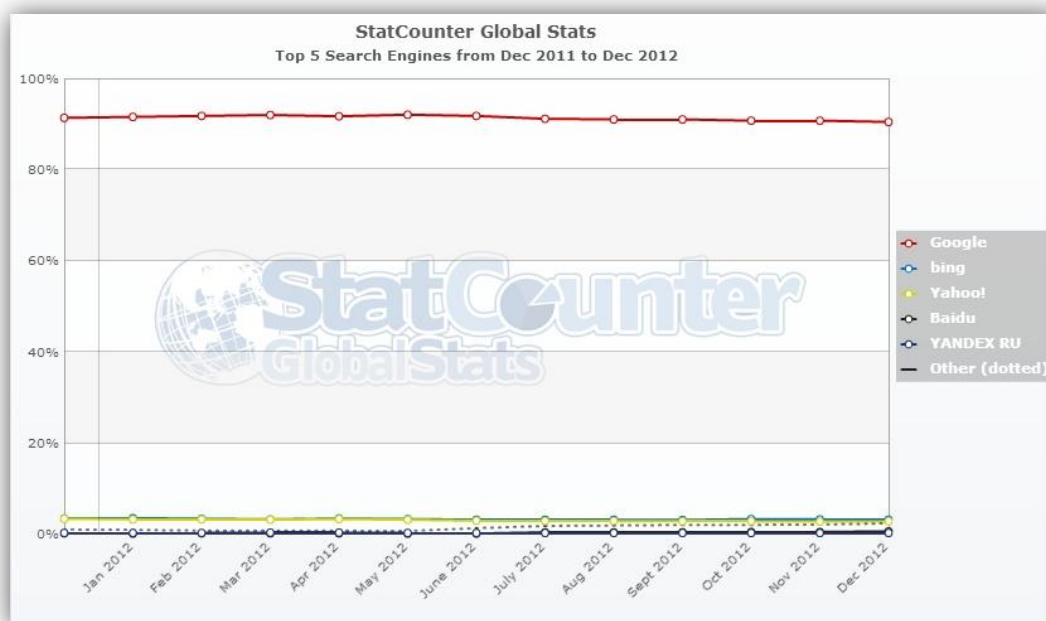


Ilustración 1. Top 5 en Motores de Búsqueda (StatCounter 2012)

¹ <http://gs.statcounter.com/>

Todos los sitios existentes en la Red desean aparecer en las primeras posiciones en los resultados de una búsqueda. Para calcular la relevancia de un sitio web que determine la posición que debe ocupar en los resultados de una búsqueda, Google utiliza un algoritmo propio denominado PageRank.

El algoritmo del PageRank fue propuesto originalmente en el documento [1] donde los fundadores Sergey Brin y Lawrence Page dieron a conocer el prototipo de Google:

$$PR(A) = (1 - d) + d \sum_{i=1}^n \frac{PR(i)}{C(i)}$$

Ecuación 1. Algoritmo PageRank inicial (1998)

Donde las variables son:

- $PR(A)$, es el PageRank de la página A .
- $PR(i)$, es el PageRank de las páginas i que enlazan A .
- $C(i)$, es el número de enlaces salientes de la página i .
- d , es un factor variable que puede estar entre 0 y 1.

Sus inventores tomaron la idea de las referencias bibliográficas que tienen los documentos académicos. Cuando un documento es referenciado un número amplio de veces por otros documentos, puede significar que el documento referenciado es relevante en esa materia.

Partiendo de esta hipótesis, hemos visto que PageRank basa principalmente su funcionamiento en los enlaces, utilizando estos como si fueran una especie referencia en un documento académico [2].

A los sitios web les será asignado un valor o PageRank en base al número de enlaces que reciben por parte de otras páginas web. Para determinar ese valor, el algoritmo también tendrá en cuenta el valor de esas páginas apuntadoras, junto con otros criterios que Google no ha hecho públicos.

En definitiva, Google utilizará los vínculos existentes en una página hacia otras como si fueran votos que realiza la página que los contiene a la página que los recibe, valorando también si la página que emite el voto es una página importante.

Para comprender mejor el funcionamiento del algoritmo, pondremos un ejemplo gráfico, donde los sitios web serán representados por bolas, en las cuales se simboliza su popularidad en función de su tamaño:

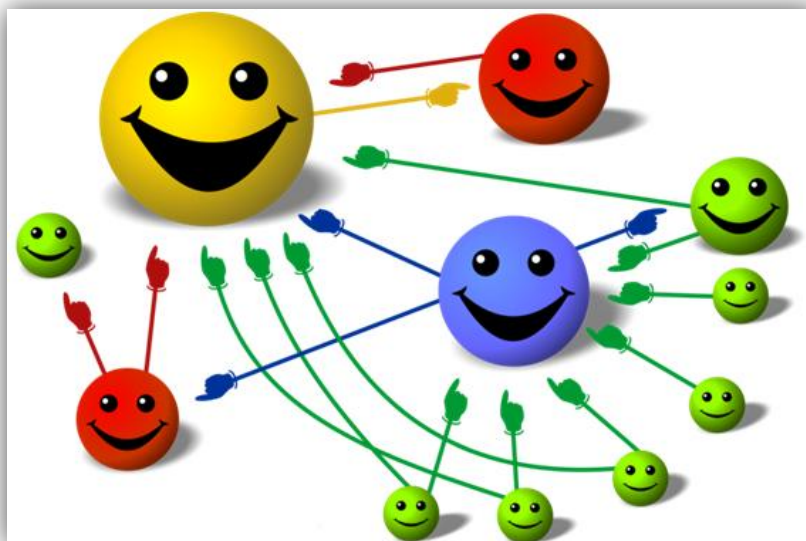


Ilustración 2. Ejemplo gráfico PageRank

A continuación mostramos un resumen de los distintos cambios que se han ido produciendo durante los últimos años en este algoritmo así como el nombre por el cual se han conocido [3]:

- *Vince* fue una apuesta de Google por mejorar los resultados de las marcas, es decir, esta actualización consistió en priorizar las marcas ante ciertos criterios de búsqueda.
- *Caffeine* se diseñó para que los resultados pudieran ser indexados de una manera más rápida. Google abordó este cambio para ofrecer resultados en tiempo real siendo capaz de mostrar el contenido que se estaba subiendo a la Red de forma instantánea.
- *MayDay* se centró en mejorar las búsquedas “Long Tail”, es decir, aquellas búsquedas formadas por cuatro o más palabras clave y que obtienen menor número de resultados.
- *Panda* provocó que se vieran penalizados los portales con poco contenido o contenido copiado, con demasiada publicidad, premiando las páginas con enlaces de calidad y no aquellas con vínculos intercambiados.

- *Freshness* supuso una modificación en el algoritmo con el objetivo de mostrar en las primeras posiciones de los resultados los sitios con contenidos más recientes relacionados con el tema de la búsqueda.

Pondremos un ejemplo para mostrar de una manera más clara los resultados que ofrece: si se realiza una búsqueda con las palabras clave “Juegos Olímpicos” Google mostrará resultados relacionados con los Juegos Olímpicos inmediatamente anteriores o futuros, dejando con menor popularidad los sitios web con información histórica sobre ellos.

- *Pingüino* fue diseñada para tratar de penalizar aquellos sitios web que utilizaran las palabras clave sobre las que tienen un interés en conseguir un buen posicionamiento, de manera repetitiva y como relleno del contenido de la web.
- *Venice* fue una actualización para modificar los resultados ofrecidos dependiendo de la localización de la búsqueda, es decir, los resultados variarán para unas mismas palabras claves según el lugar donde la consulta sea realizada.

Como ejemplo para su comprensión diremos que los resultados de una búsqueda con las palabras clave “Centro Veterinario” cambian en base a la localización para ofrecer información sobre los centros veterinarios más cercanos al lugar desde el cual se ha realizado dicha búsqueda.

2.1.1. Alternativas a PageRank

A continuación explicamos algunos de los algoritmos alternativos existentes a PageRank para calcular la popularidad de un sitio web:

2.1.1.1. HITS

HITS (Hyperlink Induced Topic Search), desarrollado por Kleinberg [4], diferencia los sitios web resultado en *hubs* o *autoridades*, a partir de las palabras clave indicadas en un motor de búsqueda.

Las páginas *autoridades* son aquellas donde la calidad de información contenida sobre el tema buscado es superior, mientras que las páginas *hubs* son las que contienen un número elevado de enlaces a páginas *autoridades*.

En la implementación del método, cada página inicialmente tiene asignado un valor $hub(H)$ y un valor $autoridad(A)$ calculadas en base al siguiente algoritmo:

$$H_T = \sum A_{ti} \text{ donde } T \rightarrow ti$$
$$A_T = \sum H_{ti} \text{ donde } ti \rightarrow T$$

Ecuación 2. Algoritmo HITS

Por tanto, el algoritmo HITS establece la valoración de una página definiendo recursivamente cada indicador a partir del otro:

- La *autoridad*, que como se ha descrito con anterioridad, valora la calidad de una página como recurso de información, usando para su cálculo la suma de los valores *hub* de los vínculos que apuntan hacia la página.
- El *hub* se halla teniendo como referencia la calidad de las *autoridades* a las que enlaza, es decir, es un indicador que dice cómo de buena es la información que se puede obtener siguiendo los enlaces contenidos en la página.

2.1.1.2. Edison

Edison fue presentado por el motor de búsqueda Ask [5]. Se trata un algoritmo de búsqueda con el que pretende competir este buscador frente al PageRank de Google.

A la hora de calcular la relevancia de un sitio web, Edison trató de proporcionar unos resultados diferentes, en los cuales se tuvieran en cuenta las opiniones de los usuarios.

Para ello el algoritmo combinó las tecnologías *Teoma* y *Direct Hit*, dos tecnologías pertenecientes a Ask. *Teoma* era una tecnología basada en enlaces, igual que las detalladas previamente, pero *Direct Hit* complementaba la puntuación de cada página contabilizando el número de clics recibidos por parte de los usuarios.

2.1.2. Valoración de los algoritmos existentes

Tanto HITS como PageRank, son algoritmos basados en enlaces, cuyo mecanismo sufre el *efecto de la contribución circular* descrito por Wang [6], basado en el hecho de que las páginas web se pueden enlazar unas a otras, de forma que se produzca un camino circular entre ellas.

Este efecto provocaría, como está expuesto en el *Planteamiento del problema* de la *INTRODUCCIÓN*, una visión estática de la Web, puesto que siempre serían las mismas páginas las que ocuparían los primeros lugares.

Una vez estudiados estos algoritmos podemos afirmar que no calculan la relevancia de una manera social y dinámica, como es nuestro objetivo.

Por su parte Edison, al tener en cuenta el tráfico de usuarios, se acerca en cierto modo a nuestra idea de calcular la relevancia de una forma social. Aun así la visión de la Web continuaría siendo estática, puesto que además de seguir teniendo en cuenta los enlaces, las páginas que se encuentren mejor posicionadas serán las que reciban más tráfico de usuarios.

2.2. Search Engine Optimization (SEO)

Debido al mecanismo de los algoritmos que hemos presentado y su conocimiento por parte de las páginas web y sus administradores, estas centran sus esfuerzos en mejorar el tráfico en su web, intentando mejorar su posicionamiento en las búsquedas.

SEO es la forma con la que se conoce al arte de preparar una web para que sea más relevante para los motores de búsqueda [7]. Generalmente la mejora del SEO se llevará a cabo en dos fases:

- SEO onPage. Es el conjunto de acciones que se realizan dentro de una página para optimizarla para los buscadores. Se debe estudiar y escoger la mejor *keyword* (palabra clave) posible para cada una de las páginas que forman el sitio web y luego orientar las páginas a dicha *keyword*.

Sobre cómo orientar las páginas hacia las *keywords* de búsqueda que nos interesa existen distintas técnicas entre las cuales podemos destacar usar títulos de página únicos, utilizar la meta-etiqueta de descripción, utilizar URL's "amigables" para hacer la página más "compartible", mejorar la navegación en el sitio, optimizar el uso de imágenes y ofrecer contenido de calidad.

- SEO offPage. Internet son relaciones entre nodos y que como hemos visto, la relevancia de cada uno viene indicada por el número de vínculos que llegan a ella y por el sitio de donde proceda este vínculo. La estrategia de SEO offPage consiste pues, en relacionar nuestra web con el resto de Internet logrando enlaces por parte de otras páginas.

En 2011, Google lanzó su servicio de búsqueda social, el botón Google+1 declarando: "Con estos cambios, queremos ayudarle a encontrar la información más relevante de la gente que te importa" [8].

Este botón significa un filtro más a tener en cuenta por los investigadores SEO que tratan de que un sitio web aparezca en los primeros resultados de una búsqueda y el cual no pueden manejar.

Así, cuando tus amigos y contactos hagan alguna búsqueda en Google, tu +1 les ayudará a encontrar lo mejor de la red. El mecanismo es el botón Google+1, que permite a los usuarios compartir páginas interesantes con sus contactos.

Podemos afirmar que el botón +1 fue un inicio para “limpiar” los resultados ofrecidos por el buscador, introduciendo la inteligencia de la persona al tener en cuenta su opinión y no únicamente mostrar resultados que pueden estar modificados porque una página ha mejorado su SEO optimizando su web o mejorando sus relaciones con otras páginas con el intercambio de enlaces.

2.3. Algoritmos sociales

En el año 2004, Tim O'Reilly bautizó el término *Web 2.0* para referirse a una nueva generación, en la cual hemos pasado de buscar información a buscar personas [9].

Inicialmente la Web fue concebida para obtener información que había sido publicada por personas con habilidades técnicas. En esta nueva era, basada en comunidades de usuarios donde cualquiera puede publicar contenido, se promueve la cooperación y el intercambio de información entre los usuarios.

En las redes sociales estamos conectados con personas con las que compartimos intereses comunes. Los usuarios de las redes sociales confían cada vez más en otros usuarios con los que tienen relación para tratar de obtener información relevante, que en los resultados proporcionados por los algoritmos de los motores de búsqueda [10].

Este hecho constata la necesidad de nuevos algoritmos en los buscadores que consideren las opiniones y acciones de los usuarios sobre un sitio web a la hora de calcular la relevancia del mismo.

Seguidamente se exponen algunos de estos algoritmos sociales donde se consideran las opiniones y/o acciones llevadas a cabo por los usuarios:

2.3.1. Collaborative Filtering

El *Collaborative Filtering* es un sistema de recomendación, con un algoritmo que tiene como objetivo presentar recomendaciones a usuarios basadas en usuarios con intereses similares.

Las listas de recomendaciones ofrecidas están elaboradas a partir de algún tipo de análisis de coincidencias. El algoritmo trabaja con técnicas estadísticas para encontrar usuarios con un historial de acciones realizadas sobre un sitio web similar al usuario actual [11].

Cuando se ha generado una lista con usuarios con intereses parecidos se combinan las preferencias de estos para construir la lista de recomendaciones que será mostrada al usuario.

Un ejemplo de utilización de éste algoritmo podemos encontrarlo en Amazon.com² una web dedicada a la venta de productos por Internet. Imaginemos un usuario que visita el portal en busca de un videojuego, si éste previamente ha realizado la compra de un videojuego en Amazon.com, el portal mostrará una lista de recomendaciones al usuario para cuya composición se ha tenido en cuenta los videojuegos adquiridos por otros clientes con los cuales el usuario ha coincidido en la compra de algún otro videojuego previamente.

En resumen, el algoritmo está basado en la suposición de que usuarios que han realizado las mismas búsquedas deben tener intereses parecidos y por tanto, las nuevas búsquedas de alguno de esos usuarios pueden resultar también interesantes para los demás.

2.3.2. EdgeRank

Como hemos visto, la Web pasó de ser una mera fuente de información a convertirse en un lugar donde los usuarios interactúan entre sí, comparten información y realizan sus propias aportaciones.

Entre esas comunidades de personas, denominadas redes sociales, donde usuarios comparten actividades e intereses, encontramos Facebook³. Esta red es la más utilizada a nivel mundial actualmente.

Los usuarios de esta red social tienen una página conocida como *muro* donde Facebook les muestra los comentarios, fotos u otras acciones del resto de usuarios con los que están relacionados.

² <http://www.amazon.com>

³ <https://www.facebook.com/>

El orden de estos resultados mostrados en el *muro* guarda relación con el tiempo, pero no es lineal, lo que nos lleva a preguntarnos cuál es el algoritmo que rige el orden en el cual los resultados son expuestos al usuario.

El algoritmo definido por Facebook, encargado de valorar la relevancia de las distintas publicaciones y decidir qué contenido es el más importante para mostrar al usuario es el EdgeRank [12]. Este algoritmo está definido en la siguiente fórmula matemática:

$$\sum_{\text{edges } e} u_e w_e d_e$$

Ecuación 3. Algoritmo EdgeRank

Donde:

- u , representa el grado de afinidad entre el usuario que realiza la publicación y el usuario que la visualiza en su *muro*. Un usuario tendrá más afinidad con otro cuando su interacción es mayor, es decir, cuando entre ellos hay un intercambio fluido de mensajes o el usuario al que se le muestra la publicación visita regularmente el perfil del usuario que la ha realizado. De esta forma el algoritmo tenderá a mostrar en los primeros lugares los contenidos creados por un usuario con afinidad que los de otro con el cual el usuario no interacciona.
- w , es el factor en el cual está representado la relevancia del contenido publicado. Para su cálculo Facebook tendrá en cuenta la interacción de todos los usuarios con la publicación, de forma que esta aumentará su valor cuanto mayor sea el número de comentarios hacia ella, mayor sea el número de clics realizados en el botón *Me Gusta* y mayor sea el número de veces que otros usuarios han compartido la publicación.
- d , es el criterio donde toma importancia el tiempo, entendiendo que cuanto mayor sea el tiempo transcurrido desde la publicación, menor importancia tendrá el contenido de la misma.

2.3.3. Particle Swarm Optimization (PSO)

Este método fue desarrollado por primera vez por Russell Eberhart y James Kennedy en 1995 [13]. Tiene su base en el comportamiento de los enjambres de insectos o bandadas de pájaros en la naturaleza y en el establecimiento de interacciones sociales entre sus individuos.

En líneas generales, la finalidad del algoritmo es encontrar el óptimo global de una función. Para ello se inspira en el movimiento de un conjunto (enjambre) de insectos (partículas).

Inicialmente el enjambre se encuentra disperso por el espacio de búsqueda, pero con el tiempo las partículas convergen hacia una zona reducida del espacio en la cual intensifican su búsqueda.

Las partículas poseen un lugar determinado dentro del enjambre, representando cada una de ellas una posible solución. Utilizan tres factores durante su búsqueda:

- *Inercia*, que viene determinada por el instinto de una partícula a continuar su búsqueda en la misma dirección.
- *Cognitivo o individual*, representando la tendencia de una partícula a reconocer lugares que históricamente le dieron un buen resultado.
- *Social*, donde se tiene en cuenta la información obtenida de otras partículas del enjambre, representando la influencia que éstas tienen sobre el individuo.

Para entender el funcionamiento del algoritmo podemos recurrir a los enjambres de abejas y el método que siguen para encontrar polen:

- Su primer objetivo será encontrar la región del espacio donde exista mayor densidad de flores, puesto que es allí donde podrán encontrar mayor cantidad de polen.
- El enjambre la única información de la que dispone es histórica, reconociendo la región del espacio donde en el pasado encontró una mayor densidad de flores.
- Una abeja por instinto podrá variar su movimiento. En caso de que ésta abeja localice un lugar con mayor densidad de flores del que tenían conocimiento hasta ese momento, el enjambre modificará su exploración dirigiéndose hacia esa nueva región.

2.4. Ant Colony Optimization (ACO)

Para lograr alcanzar los *Objetivos* de éste Trabajo Fin de Grado nos hemos centrado en el campo de la investigación ACO (Ant Colony Optimization) sobre la optimización basada en colonias de hormigas.

Las hormigas viven en sociedad, son insectos sociales, y su comportamiento va encaminado a la supervivencia de ésta. La alimentación y su enfoque para encontrar la ruta más corta entre su hormiguero y la fuente de alimento [14], provoca que los investigadores encuentren una conducta muy importante para sus estudios.

2.4.1. Hormigas naturales

Cuando las hormigas salen en busca de comida, exploran los alrededores en todas direcciones de forma un tanto aleatoria. Si una hormiga encuentra algo de interés, volverá a la colonia depositando feromonas en el camino, una sustancia química que las hormigas son capaces de detectar. Esta señal define el camino entre la colonia y la comida.

Las hormigas depositarán mayor cantidad de feromonas cuando la cantidad y la calidad de la comida sean más altas. Esta sustancia será utilizada por el resto de “compañeras” de la colonia para seguir el rastro dejado por la primera y poder llegar a la fuente de alimento.

Cuando la primera hormiga llega al nido informado de que ha encontrado alimento. En el hormiguero el resto descubre la ruta de feromona depositada y deciden seguirla. Al llegar al alimento, éstas estiman, al igual que hizo la primera, la cantidad y calidad del mismo, segregando más feromonas sobre el camino. De esta forma, si el alimento es apropiado para la colonia, rápidamente se acumula un número elevado de hormigas en la fuente de alimento.

Siendo animales prácticamente ciegos, llama la atención como las hormigas además de ser capaces de encontrar comida y de comunicarse mediante las feromonas para que los miembros de la colonia consigan llegar al alimento, tienen la capacidad de encontrar el camino más corto entre su nido y la fuente de alimento, demostrando su alto grado de estructuración.

Éste último comportamiento descrito fue descubierto por Jean-Louis Deneubourg, investigador de la Universidad de Bruselas [15].

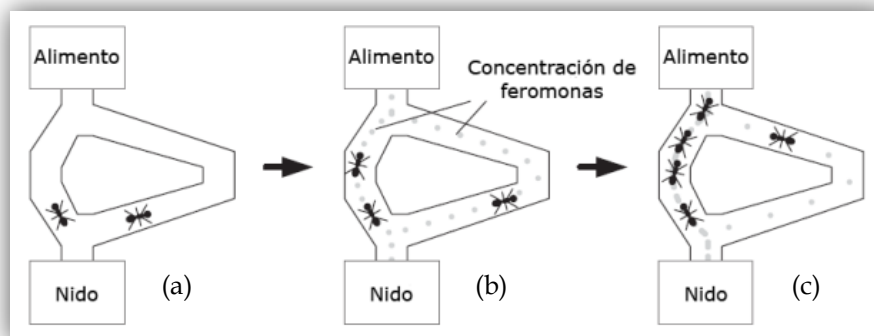


Ilustración 3. Comportamiento de las hormigas en busca de una ruta al alimento

En la ilustración anterior se puede observar como las hormigas terminan trazando la ruta más corta hacia la comida. ¿Cómo es posible?

- En la primera de las figuras (a) las hormigas salen del nido en busca de comida, exploran aleatoriamente la región del espacio y al no existir ningún rastro de feromonas, cuando encuentran una bifurcación en su camino decide la ruta para continuar de un modo probabilístico. Por ello en la figura vemos como el 50% de las hormigas elige un camino y el otro 50% elige el otro.
- Como hemos visto las hormigas desprenden una sustancia química que el resto de hormigas seguirán para llegar a la fuente de alimento. Cuando ambas hormigas regresan al nido indicando que han encontrado una fuente de alimento el resto de hormigas de la colonia sale siguiendo el rastro en busca del alimento.
- En la figura (b) se empieza a observar que la concentración de feromonas en la ruta más corta es mayor, lo que es debido a que las hormigas que utilizan este camino, regresarán con anterioridad al nido que las que utilizan el camino más largo.
- Por ello en la figura (c) se ve claramente como las nuevas hormigas que salen del nido tienen una mayor probabilidad de seguir la ruta con mayor concentración de feromonas, la cual representa el camino más corto. De esta forma individuos tan simples como pueden ser las hormigas, trabajando conjuntamente llegan a optimizar la ruta existente entre el hormiguero y la comida.

Estos marcadores químicos son evaporados con el paso del tiempo, de ésta forma los senderos sin interés y sin usar desaparecerán de manera progresiva. Esta evaporación conlleva a que las hormigas necesiten explorar nuevamente las regiones en busca de comida.

2.4.1. Hormigas virtuales

La investigación ACO tiene su punto de partida en éste comportamiento de las colonias reales de hormigas. El primer algoritmo inspirado en esa investigación fue *Ant System (AS)* propuesto por Marco Dorigo y sus colaboradores con la descripción del *Problema del Vendedor Viajero* [16] en 1996.

Este problema consiste en que dado un conjunto ciudades finito y el coste del viaje entre todos los pares de ciudades, el viajante debe encontrar la manera de visitar todas las ciudades exactamente una vez y regresar a la ciudad de partida de la forma más barata.

Tanto las hormigas virtuales como las hormigas naturales forman conjuntos de individuos muy simples que trabajan para alcanzar su objetivo de una manera conjunta. Una de las hormigas puede alcanzar la solución, pero es la cooperación de todas la que conduce a su optimización.

Las tres ideas principales tomadas para modelar en el entorno computacional el comportamiento de las hormigas reales fueron:

- Las feromonas serán utilizadas por las hormigas virtuales para seguir el rastro trazado por otras hormigas virtuales. Su representación será mediante un tipo de información numérica.
- Las rutas más cortas tienden a tener un valor de feromona más alto.
- La elección de un camino por parte de las hormigas será una decisión probabilista mostrando preferencia por aquellas rutas con un valor de feromona más alto.

Además se aportaron a las hormigas artificiales capacidades que no poseen las hormigas naturales. Seguidamente citamos cuales fueron estas capacidades:

- Una hormiga será capaz de determinar la distancia existente a la que se encuentra un objetivo, de forma que pueda utilizarlo en la toma de decisiones. Relacionándolo con el problema del viajero cuando un usuario llega a una bifurcación, en la cual cada camino le conduce a una ciudad, la hormiga podrá utilizar esta información para elegir el camino que desea seguir.
- Las hormigas tendrán memoria, asegurando así que únicamente se obtengan soluciones posibles. En el problema del viajero, éste solamente podrá visitar cada ciudad en una ocasión, si las hormigas tienen memoria serán capaces de determinar si una ciudad la han visitado con anterioridad.

Existen varios algoritmos ACO, pero todos se basan en el algoritmo AS descrito anteriormente en líneas generales y sobre el cual se realizan pequeñas modificaciones con el propósito de disminuir el error.

Entrando en más detalle, en ACO todos los algoritmos se basan en un modelo donde cada hormiga virtual es un elemento que trabaja en la construcción de una solución al problema de una forma probabilística. El problema debe poder representarse en forma de grafo con pesos, donde cada arco (a) del grafo contendrá:

- Rastros artificiales de feromona (τ) que varían con el transcurso del tiempo reflejando la experiencia recogida por las hormigas durante la búsqueda de la solución.
- Información heurística (η) que depende del caso concreto del problema y no es modificada durante la ejecución del algoritmo. En este caso la información heurística empleada sería la distancia entre nodos del arco.

Para realizar la elección del movimiento hacia un nodo la hormiga artificial usa un mecanismo que:

- Almacena en una lista (L) los nodos recorridos, para no repetir visitas. De esta forma cuando la hormiga finalice su camino, la lista contendrá la solución construida.
- Encontrándose en el nodo r es capaz de elegir entre los nodos vecinos hacia qué nodo s dirigirse ($N(r) = \{s \mid \exists a_{rs} \text{ y } s \notin L\}$) utilizando la regla probabilística de transición siguiente:

$$p_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^\alpha \times [\eta_{rs}]^\beta}{\sum_{u \in N_r^k} [\tau_{ru}]^\alpha \times [\eta_{ru}]^\beta}, & \text{si } s \in N_k(r) \\ 0, & \text{En otro caso} \end{cases}$$

Ecuación 4. Regla probabilística de transición (ACO)

Donde:

- ❖ $N_k(r)$ es el conjunto de ciudades o nodos alcanzables por la hormiga k desde r que aún no se han visitado.
- ❖ τ_{rs} , representa el rastro de feromona entre los puntos r y s .
- ❖ η_{rs} representa el valor de la información heurística del arco a_{rs} .
- ❖ α y β son parámetros que establecen un equilibrio entre la importancia de la heurística utilizada y los valores de feromona detectados. Sus valores deben estar contenidos en el intervalo $[0, 1]$.

La definición de estos valores puede modificar notablemente el resultado obtenido. En caso de tomar $\alpha = 0$, únicamente se estaría teniendo en consideración la función heurística elegida. Por el contrario si tomamos el valor $\beta = 0$ el único dato tenido en cuenta serán los rastros de feromona.

Otro aspecto importante en ACO es la actualización de la feromona que se produce cuando la hormiga alcanza su solución con el objetivo de incrementar los valores de feromona asociados a soluciones buenas o prometedoras, disminuyendo el valor en las malas soluciones. Para este mecanismo se deben tener en cuenta dos subprocesos:

- *Aporte de feromona.* Cuanto mejor sea la solución obtenida, mayor cantidad de feromona será depositada.
- *Evaporación de la feromona.* Es utilizada para eliminar las decisiones tomadas erróneamente y evitar un incremento ilimitado de los rastros de feromona existentes.

La actualización de la feromona en el modelo tiene la siguiente expresión:

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs} + \sum_{k=1}^m \Delta \tau_{rs}^k$$

Ecuación 5. Actualización de la feromona (ACO)

Donde:

- ρ es el parámetro de evaporación de feromona ($\rho \in [0, 1]$).
- m es número de hormigas.
- $\Delta \tau_{rs}^k = \begin{cases} \frac{1}{C(S_k)}, & \text{si la hormiga } k \text{ ha visitado el arco } a_{rs}. \\ 0, & \text{en otro caso.} \end{cases}$
- $C(S_k)$ es el coste de la solución obtenida por la hormiga k , siendo éste la longitud del circuito S_k .

Con este proceso las hormigas depositan feromona extra en los arcos del grafo que forman parte de la solución, por considerarlos arcos prometedores, consiguiendo que las mejores soluciones tengan un mayor refuerzo. Por el contrario en aquellos arcos que no han sido visitados por ninguna hormiga y por lo tanto no forman parte de la solución sufren la evaporación de un porcentaje de la feromona que disponían.

2.5. Comparación PageRank - ACO

Para finalizar este capítulo queremos comparar, en una tabla a modo resumen, el algoritmo PageRank utilizado por Google y la propuesta basada en ACO sobre la que estamos trabajando.

	PageRank	ACO
Visión de la Web	Estática	Dinámica
Cálculo de relevancia	Enlaces	Social

Tabla 1. Comparación PageRank - ACO

Pese a que el algoritmo de Google puede satisfacer a muchos usuarios, pensamos que tiene la desventaja de estar basado en una visión estática de la web, según la cual los sitios importantes serán cada vez más importantes. Esto es debido, como hemos visto, a que la relevancia del sitio es calculada en función al número de enlaces existentes sobre él en distintas páginas.

ACO nos ofrece una nueva visión a la hora de calcular la relevancia de un sitio web. Considerando a los usuarios como “hormigas virtuales”, el cálculo de la relevancia estará basado en el interés causado por un sitio en los usuarios consiguiendo que los resultados tengan un carácter social y dinámico.

3. ANÁLISIS DE LA PROPUESTA

En el presente capítulo se pretende mostrar la solución propuesta para lograr un recurso que cumpla con los *Objetivos* planteados.

3.1. Propuesta

En Internet una de las ocupaciones a las que los usuarios dedican mayor cantidad de tiempo es en buscar información de su interés. Por ello uno de los temas más interesantes que pueden ser estudiados son las estrategias a seguir para mejorar la obtención de esta información.

Con la teoría del rastreo de información o *Information Foraging* desarrollada por Pirolli y Card [17] se percataron que nosotros (los humanos) cuando tratamos de encontrar información utilizamos técnicas similares a las usadas por animales en busca de alimento.

Un concepto clave en la búsqueda de información es *Information Scent* que está basado en los mecanismos que tienen los animales cazadores para identificar las rutas más adecuadas que llevan hasta su presa. Si existe un rastro indicador que aporte ideas sobre la fuente de alimento que lo produjo, un cazador recolector tendrá más información a la hora de elegir un camino.

Extrapolando este concepto al entorno web un usuario que trata de buscar una página de su interés, simplemente con el enlace puede tener conocimiento del contenido de ese sitio. El rastro estaría producido por la fuente de información, dando una idea al usuario de la información que se puede obtener tras un enlace.

Se quiere que los resultados mostrados tras una búsqueda por parte de un usuario tengan un carácter social y dinámico. Se pretende calcular su relevancia para la ordenación en el ranking de resultados en base a las acciones realizadas por otros usuarios que realizaron una misma búsqueda.

3.1.1. Herramienta propuesta basada en ACO

Para conseguir nuestro propósito se ha tomado la idea de la biología, centrándonos en el campo de la investigación ACO: la forma de buscar el alimento que tienen las hormigas.

Éste consiste en modelar el comportamiento de los usuarios como si fueran “hormigas virtuales” en busca de un espacio de soluciones (“comida”) mediante el uso de rastros dejados por otros [18].

Supongamos un conjunto de usuarios que realiza la misma consulta en un motor de búsqueda, como puede ser un mismo conjunto de palabras clave. Sus acciones posteriores pueden ser modeladas como si estuvieran construyendo una solución a su problema de búsqueda de información. Nosotros modelamos esta situación con senderos virtuales [19].

La elección tomada por los usuarios cuando cliquean un enlace u otro puede llevarle directamente a su objetivo o por el contrario puede conducirle a una información falsa, por lo que éste retornará al motor de búsqueda para explorar nuevos enlaces.

Al igual que las hormigas en busca de alimento, los usuarios exploran aleatoriamente los resultados ofrecidos por el motor de búsqueda e irán depositando un peso o valor a cada sitio visitado en lo que consideraríamos una especie de feromonas.

A su vez, dependiendo de las acciones de estos sobre dicho sitio web, cuando éste contenga una gran cantidad o calidad de información, el peso depositado podrá aumentar. Haciendo un símil con el incremento de feromonas cuando una hormiga considera que la fuente de alimento tiene una gran cantidad de comida o que la calidad de ésta es muy buena.

Estas acciones que los usuarios pueden llevar a cabo y las cuales se pretenden tener en cuenta para la construcción de los senderos virtuales son aquellas donde el usuario pulsa un botón “me gusta”, pasa un tiempo determinado en una página o añade ese sitio a los favoritos de su navegador.

Por último se pretende que cuando una fuente de información queda obsoleta, ésta vaya evaporando sus feromonas, vaya perdiendo su peso en el recorrido.

De esta forma, en nuestro modelo de sendero virtual, asumimos que si a los usuarios se les proporciona un rastro de información adecuada seguirán la siguiente meta-heurística:

```
Procedure VirtualTrails metaheuristic
    ScheduledActivities
        ShowVirtualTrails()
        ManageUserActivity()
        EvaporatePheromone()
    End ScheduledActivities
End VirtualTrails metaheuristic
```

En la meta-heurística la primera función tiene como objetivo mostrar a los usuarios pistas realizadas por usuarios anteriores los cuales han ido depositando feromonas. Su segundo paso consiste en añadir feromonas a los enlaces en relación a la actividad llevada a cabo por los usuarios. Y por último se actualizará la evaporación de las feromonas.

Basándonos en la meta-heurística anterior, para conseguir la primera de las funciones, `ShowVirtualTrails()`, el sistema almacenará en una base de datos las búsquedas realizadas por los usuarios en el buscador y su recorrido posterior a través de los distintos sitios web en busca de información de su interés.

De ésta forma cuando un usuario realice una búsqueda, la herramienta a su vez consultará en su base de datos si ha sido realizada previamente por otros usuarios. Y así podrá mostrar una lista de enlaces alternativa a los resultados dados por el buscador. Esta lista de enlaces ofrecida por la herramienta tendrá la característica de estar basada en sitios web en los cuales los usuarios anteriores que realizaron la misma consulta encontraron la información que buscaban.

Para poder mostrar la lista de enlaces, según la función `ManageUserActivity()`, la herramienta otorgará un peso a las páginas web visitadas por el usuario en su recorrido tras una búsqueda. El peso servirá para clasificar la página y variará dependiendo de las acciones que realice el usuario sobre ella.

Cuando el usuario visite una página se comprobará si ésta pertenece al recorrido almacenado en la base de datos de forma que pueda añadirse al mismo o por el contrario aumentar el peso de dicha página en el recorrido existente.

No siempre se llega al sitio web deseado a través de un enlace ofrecido por el motor de búsqueda, sino que otras muchas veces para lograr llegar a una página interesante el usuario ha tenido que ir navegando a través de distintos sitios web. Por ello cuando se visita una página se comprobará si pertenece a un recorrido que ha podido seguir el usuario desde su consulta en el buscador. Para que una página pertenezca al recorrido se debe haber llegado a ella a través de un enlace de la página de resultados del buscador o con origen en dichos resultados.

Una vez es comprobado que la página pertenece al recorrido seguido por un usuario tras su consulta en el motor de búsqueda, se tendrá en cuenta tanto el tiempo de permanencia en la página como las acciones que pueda llevar a cabo el usuario sobre ella y que sean indicativas de que el sitio ha resultado interesante para su búsqueda.

Así el peso asignado a una página tendrá un carácter social puesto que estará basado en acciones que indican que el sitio web ha resultado interesante para el usuario.

En el modelo de actualización de la feromona, cada vez que sea visitada una página (u) perteneciente al recorrido (r), el aporte de feromona será realizado en base a la siguiente regla:

$$\tau_u(r) = \tau_u + 1 + (1 \cdot m) + \sum_{k=0}^n a$$

Ecuación 6. Aporte de feromona (Virtual Ants)

Donde:

- τ_u el peso de la URL en el recorrido.
- 1 es el valor de feromona depositado por visitar la página.
- m es el número de minutos pasados en la página. Se aporta un valor de feromona igual a 1 por cada minuto que el usuario permanezca en ella.
- n es el número de acciones. El conjunto de acciones indicativas de interés contempladas son: añadir la URL a favoritos, hacer clic en los botones de redes sociales que se encuentren en la página e imprimir la página.
- a es el peso depositado si el usuario realiza una acción k sobre el sitio.

También conocemos que una web que forma parte de una solución ha podido quedar obsoleta y dejar de ser interesante para los usuarios, como puede suceder por una falta de actualización de su contenido o que éste pertenezca a una noticia antigua, en cuyo caso necesitaremos que vaya disminuyendo progresivamente su peso en el recorrido para favorecer a las web que resulten de interés en el momento en el que se realiza la consulta.

Para ésta última función, `EvaporatePheromone()`, la herramienta comprobará la última visita o acción realizada sobre los sitios web almacenados en los distintos recorridos de las consultas, haciendo que los sitios desactualizados vayan reduciendo su relevancia.

La función de evaporación de una URL (u) en un recorrido (r) en el modelo de actualización de feromona propuesto tiene la siguiente expresión:

$$\tau_u(r) = (1 - \rho) \cdot \tau_u$$

Ecuación 7. Evaporación de feromona (Virtual Ants)

Donde:

- τ_u el peso de la URL en el recorrido.
- ρ es el parámetro de evaporación de feromona ($\rho \in [0, 1]$).

3.1.2. Plataforma propuesta

La solución consiste en implementar una herramienta que ofrezca a los usuarios una respuesta alternativa cuando estos realizan una consulta en el motor de búsqueda Google.

Nuestra propuesta será crear una extensión para Google Chrome⁴. Este navegador permite su personalización por parte del usuario instalando extensiones, temas o aplicaciones web.

Las extensiones permiten añadir nuevas funciones al navegador. Por ejemplo, una extensión de notificación de mensajes de correo electrónico puede mostrar una alerta en la barra de herramientas del navegador para no tener que acceder a tu correo y comprobar si hay nuevos mensajes.

También hay otras extensiones que se ejecutan en segundo plano. Estas extensiones son pequeños programas que añaden funciones útiles a Google Chrome, como la aplicación automática de formato a las páginas web y su presentación en el estilo elegido [20].

Éstas últimas son en las cuales nos queremos centrar, puesto que nuestro objetivo es ofrecer de forma automática, cuando un usuario realice una búsqueda por palabras clave en Google, unos resultados alternativos a los dados por el buscador.

También necesitamos que la herramienta trabaje en el resto de páginas web visitadas a partir de una búsqueda en Google para ir recabando toda la información sobre las rutas seguidas por los usuarios en busca de su objetivo. Y este trabajo debe realizarse en un segundo plano.

Además, Google Chrome, es el navegador más aventajado a la hora de hacer una búsqueda web, solo es necesario escribir la palabra o termino de búsqueda en la barra de direcciones y directamente te muestra los resultados ofrecidos por el motor de búsqueda Google para las palabras clave introducidas.

⁴ Google Chrome es un navegador web desarrollado por Google.

En resumen, proponemos crear una extensión para el navegador Google Chrome, que modifique la página donde Google muestra sus resultados para añadir junto a ellos un ranking alternativo donde se mostrarían los enlaces obtenidos para las mismas palabras clave en base a la investigación que estamos llevando a cabo.

3.2. Condiciones de entorno

Para la utilización de la herramienta es necesario que los usuarios dispongan de un dispositivo con conexión a Internet y el navegador web Google Chrome versión 4⁵ o superior, para el cual se va a desarrollar la extensión, instalado.

3.3. Especificación de requisitos

Seguidamente exponemos los requisitos obtenidos tras el análisis del sistema realizado. Existen dos niveles de requisitos: requisitos de usuario y requisitos de software.

Se seguirán las normas contenidas en el *ESA software engineering standards (PSS-05-0)* [21] para la definición de los requisitos.

3.3.1. Requisitos de usuario

Son las declaraciones de los servicios que debe prestar el sistema, así como las restricciones bajo las que debe operar, es decir, recogen las características más generales de la herramienta.

Previamente a integrar los requisitos vamos a especificar cuál es el método que hemos seguido para establecer su identificador:



UT-nnn

Ilustración 4. Nomenclatura requisitos de usuario

Donde:

- U – Indica que se trata de un requisito de usuario.
- T – Indica qué tipo requisito representa, admitiendo dos valores:
 - ❖ C – Requisito de capacidad.
 - ❖ R – Requisito de restricción.

⁵ A partir Google Chrome versión 4, este navegador incorpora las extensiones.

- nnn – Es el número identificativo del requisito, dentro del tipo de requisito al que pertenecen.

Dentro de cada requisito éste tiene representados unos atributos. A continuación detallamos su significado:

- *Identificación*: cada requisito incluirá una identificación, según la nomenclatura detallada, para facilitar su traza en fases sucesivas.
- *Necesidad*: indica el grado de exigencia del requisito. Puede tener tres valores: alta (cuando el requisito es esencial), media (cuando el requisito es deseable incluirlo) y baja (cuando es opcional). Los requisitos esenciales no son negociables.
- *Prioridad*: cada requisito incluirá una medida de prioridad para que el desarrollador pueda decidir la planificación de la producción.
- *Estabilidad*: algunos requisitos se pueden saber fijos sobre la vida esperada del software, mientras que otros pueden depender de las decisiones tomadas.
- *Descripción*: descripción del requisito.

3.3.1.1. Requisitos de capacidad

Especifican la funcionalidad que se desea que cumpla el sistema.

UC-001			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	El usuario podrá añadir la herramienta al navegador Google Chrome.		

Tabla 2. UC-001

UC-002			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta modificará la pantalla en la que Google muestra sus resultados.		

Tabla 3. UC-002

UC-003			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta mostrará un ranking alternativo de resultados al ofrecido por Google.		

Tabla 4. UC-003

UC-004			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta aplicará un valor a los sitios web visitados a partir de una búsqueda realizada. Los pesos variarán en función de la acción realizada por el usuario.		

Tabla 5. UC-004

UC-005			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta disminuirá el valor de una URL transcurrido un tiempo estimado sin que se produzca una acción sobre ella.		

Tabla 6. UC-005

3.3.1.2. Requisitos de restricción

Especifican la forma en que el sistema debe alcanzar los objetivos o realizar las funcionalidades.

UR-001			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta estará diseñada para cumplir su funcionalidad en la versión española del motor de búsqueda Google (https://www.google.es/).		

Tabla 7. UR-001

UR-002			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Para la utilización de la herramienta es obligatorio el registro previo en ella.		

Tabla 8. UR-002

UR-003			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	El usuario tendrá que iniciar sesión para utilizar la herramienta.		

Tabla 9. UR-003

UR-004			
Necesidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Estabilidad:	Debe existir previamente a que la herramienta sea puesta a disposición de los usuarios.		
Descripción:	Deberá existir un manual de usuario para detallar cómo se utiliza la herramienta.		

Tabla 10. UR-004

UR-005			
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta deberá cumplir la LOPD (Ley Orgánica de Protección de Datos) [22].		

Tabla 11. UR-005

3.3.2. Casos de uso

Los casos de uso describen todas las interacciones que tendrán lugar entre la herramienta y el usuario. Se extraen de los requisitos de usuario y servirán como base para los requisitos de software. A continuación mostramos un diagrama con los casos de uso considerados para nuestra aplicación:

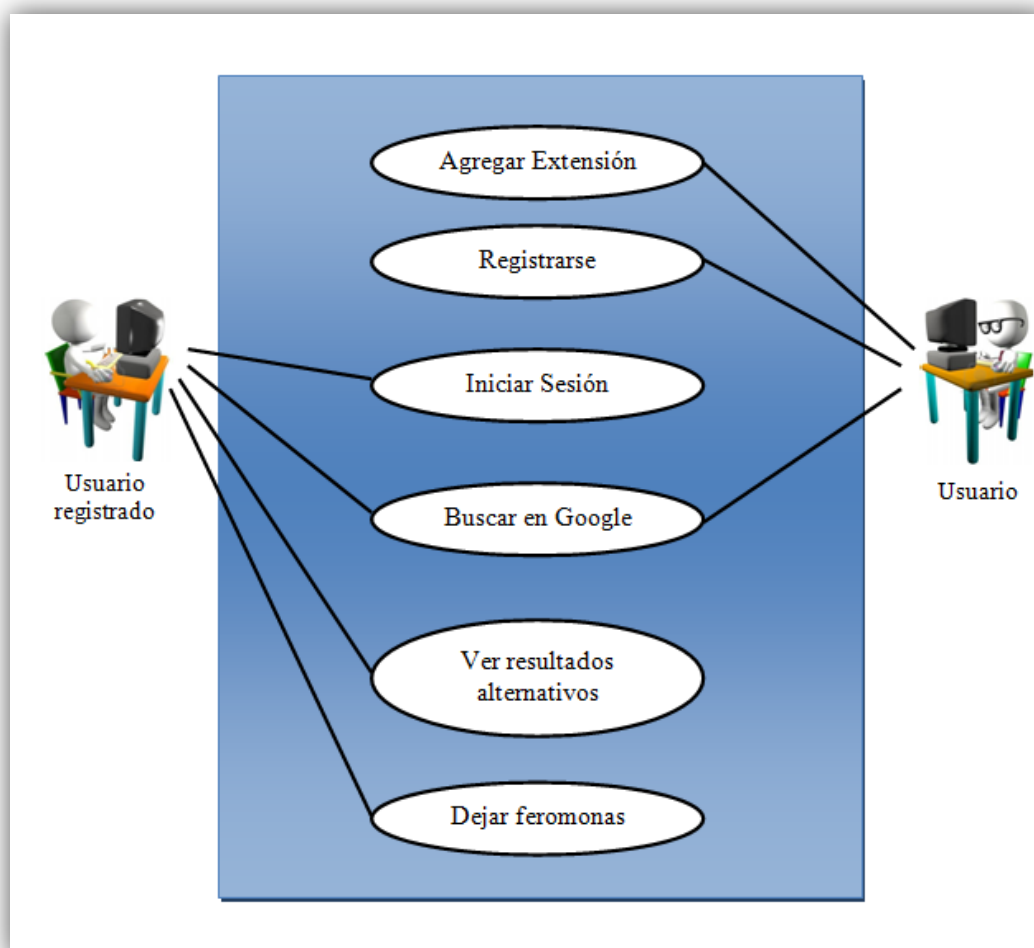


Ilustración 5. Diagrama de casos de uso

La interacción “Buscar en Google” es realizada por ambos actores de forma independiente, es decir, la acción no es llevada a cabo de manera conjunta.

Seguidamente mostramos un esquema detallando la funcionalidad de cada uno de los casos de uso. La precondition está formada por el conjunto de condiciones que se tienen que cumplir para que se pueda iniciar un caso de uso. En muchos casos supone la ejecución de casos de uso previos. La postcondición refleja el estado en que se queda el sistema una vez ejecutado el caso de uso.

Agregar Extensión	
Actor:	Usuario
Objetivo:	Disponer de la herramienta
Precondiciones:	Tener instalado Google Chrome
Postcondiciones:	Poder hacer uso de la herramienta

Tabla 12. Caso de uso - Agregar Extensión

Registrarse	
Actor:	Usuario
Objetivo:	Hacer uso de la herramienta
Precondiciones:	Tener agregada la extensión
Postcondiciones:	Poder iniciar sesión

Tabla 13. Caso de uso - Registrarse

Iniciar Sesión	
Actor:	Usuario registrado
Objetivo:	Hacer uso de la herramienta
Precondiciones:	Estar registrado en el sistema
Postcondiciones:	Poder acceder a los resultados ofrecidos por la herramienta en una búsqueda de Google

Tabla 14. Caso de uso - Iniciar Sesión

Buscar en Google	
Actor:	Usuario y Usuario registrado
Objetivo:	Buscar enlaces a sitios web de interés
Precondiciones:	Elegir la <i>keyword</i> (palabra clave) de la búsqueda
Postcondiciones:	Disponer de unos vínculos a sitios web que pueden resultar interesantes

Tabla 15. Caso de uso - Buscar en Google

Ver resultados alternativos	
Actor:	Usuario registrado
Objetivo:	Disponer de enlaces a sitios web que hayan resultado de interés para otros usuarios que realizaron una búsqueda con la misma <i>keyword</i>
Precondiciones:	Realizar una búsqueda en Google
Postcondiciones:	Disponer de unos vínculos a sitios web que pueden resultar interesantes

Tabla 16. Caso de uso - Ver resultados alternativos

Dejar feromonas	
Actor:	Usuario registrado
Objetivo:	Dejar información sobre los caminos seguidos en busca de un sitio web de interés
Precondiciones:	Realizar una búsqueda en Google
Postcondiciones:	Disponer de los enlaces que han resultado de interés para el usuario

Tabla 17. Caso de uso - Dejar feromonas

3.3.3. Requisitos de software

La especificación de estos requisitos es una descripción completa del comportamiento del sistema que se va a desarrollar.

Previamente a integrar los requisitos vamos a especificar cuál es el método que hemos seguido para establecer su identificador:

ST-nnn

Ilustración 6. Nomenclatura requisitos de software

Donde:

- S – Indica que se trata de un requisito de software.
- T – Indica qué tipo requisito representa, admitiendo los siguientes valores:
 - ❖ F – Requisitos de funcionalidad.
 - ❖ I – Requisitos de interfaz.
 - ❖ O – Requisitos de operación.
 - ❖ R – Requisitos de recursos.
 - ❖ C – Requisitos de comprobación.
 - ❖ A – Requisitos de aceptación de las pruebas.
 - ❖ D – Requisitos de documentación
 - ❖ S – Requisitos de seguridad.
- nnn – Es el número identificativo del requisito, dentro del tipo de requisito al que pertenecen.

Dentro de cada requisito éste tiene representados unos atributos. A continuación detallamos su significado:

- *Identificación*: cada requisito incluirá una identificación, según la nomenclatura detallada, para facilitar su traza en fases sucesivas.
- *Fuente*: indica el origen del requisito.
- *Necesidad*: indica el grado de exigencia del requisito. Puede tener tres valores: alta (cuando el requisito es esencial), media (cuando el requisito es deseable incluirlo) y baja (cuando es opcional). Los requisitos esenciales no son negociables.
- *Prioridad*: cada requisito de software incluirá una medida de la prioridad para que el desarrollador pueda decidir la planificación de la producción.
- *Estabilidad*: algunos requisitos se pueden saber fijos sobre la vida esperada del software, mientras que otros pueden depender de las decisiones tomadas.
- *Descripción*: descripción del requisito.

3.3.3.1. Requisitos de funcionalidad

Especifican qué tiene que hacer el software.

SF-001			
Fuente:	UC-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Debe ser posible agregar la herramienta como una extensión a Google Chrome.		

Tabla 18. SF-001

SF-002			
Fuente:	UR-002		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta permitirá realizar el registro de un usuario.		

Tabla 19. SF-002

SF-003			
Fuente:	UR-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta permitirá a los usuarios registrados iniciar sesión.		

Tabla 20. SF-003

SF-004			
Fuente:	UC-002, UC-003, UR-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta comprobará si el usuario se encuentra en el motor de búsqueda Google.		

Tabla 21. SF-004

SF-005			
Fuente:	UC-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta realizará una consulta a su base de datos cuando un usuario realice una búsqueda.		

Tabla 22. SF-005

SF-006			
Fuente:	UC-003, UC-004		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta ordenará, en función de su peso, los enlaces que han resultado de interés para otros usuarios que realizaron la misma búsqueda.		

Tabla 23. SF-006

SF-007			
Fuente:	UC-002, UC-003, UR-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta mostrará un ranking de resultados, alternativo al ofrecido por Google, con los sitios web que resultaron de interés para otros usuarios que realizaron la misma búsqueda.		

Tabla 24. SF-007

SF-008			
Fuente:	UC-004, UR-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta almacenará los caminos seguidos por los usuarios tras realizar una búsqueda en Google.		

Tabla 25. SF-008

SF-009			
Fuente:	UC-004		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta almacenará información del sitio web visitado para su visualización en Google.		

Tabla 26. SF-009

SF-010			
Fuente:	UC-004		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta capturará distintas acciones de los usuarios sobre los diferentes sitios web.		

Tabla 27. SF-010

SF-011			
Fuente:	UC-005		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta disminuirá el peso o valor de los enlaces cuando estos lleven un tiempo sin ser actualizados.		

Tabla 28. SF-011

3.3.3.2. Requisitos de interfaz

Especifican hardware y/o software con el que el sistema o componentes del sistema deben interactuar o comunicarse.

SI-001			
Fuente:	UR-002		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Existirá un pop-up (ventana emergente) en el navegador que permita el registro de usuarios.		

Tabla 29. SI-001

SI-002			
Fuente:	UR-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Existirá un pop-up (ventana emergente) para el inicio de sesión.		

Tabla 30. SI-002

SI-003			
Fuente:	UR-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Existirá un pop-up (ventana emergente) en el que comprobar si la sesión está iniciada.		

Tabla 31. SI-003

SI-004			
Fuente:	UR-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Existirá un pop-up (ventana emergente) para que el usuario pueda cerrar su sesión.		

Tabla 32. SI-004

SI-005			
Fuente:	UR-002		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Existirá un pop-up (ventana emergente) para que el usuario pueda borrar su usuario de la base de datos de la aplicación.		

Tabla 33. SI-005

SI-006			
Fuente:	UC-002, UR-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta modificará el DOM ⁶ de la página de resultados de Google para mostrar los resultados ofrecidos por nuestro sistema.		

Tabla 34. SI-006

⁶ El DOM es la estructura de objetos que genera el navegador cuando se carga un documento (página web).

3.3.3.3. Requisitos de operación

Especifican aquellos requisitos que van a indicar cómo va a realizar el sistema las tareas para las que ha sido construido.

SO-001			
Fuente:	UR-002		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Se creará una tabla en la base de datos para almacenar los usuarios registrados.		

Tabla 35. SO-001

SO-002			
Fuente:	UR-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Cuando un usuario inicie sesión se comprobará si existe en nuestra base de datos de usuarios registrados. En caso negativo se le redirigirá a la página de registro.		

Tabla 36. SO-002

SO-003			
Fuente:	UR-003, UC-003, UR-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Cuando un usuario realice una búsqueda en Google se comprobará si el usuario tiene sesión iniciada.		

Tabla 37. SO-003

SO-004			
Fuente:	UC-003, UC-004		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Cuando un usuario visite un sitio web se comprobará si esa página pertenece a la ruta. Si se ha llegado al sitio tras una consulta en Google.		

Tabla 38. SO-004

SO-005			
Fuente:	UC-006		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	El envío/recepción de información a la base de datos será realizado mediante peticiones AJAX.		

Tabla 39. SO-005

3.3.3.4. Requisitos de recursos

Especifican los límites en recursos.

SR-001			
Fuente:	UC-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	La herramienta deberá añadirse a un navegador Google Chrome, mínimo versión 4.		

Tabla 40. SR-001

SR-002			
Fuente:	UR-002, UC-004		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Se utilizará una base de datos MySQL 5.5.27 para almacenar la información de la herramienta.		

Tabla 41. SR-002

3.3.3.5. Requisitos de comprobación

Especifican cómo el software debe validar los datos de entrada y salida.

SC-001			
Fuente:	UR-002		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Para registrarse será necesaria una dirección de email válida.		

Tabla 42. SC-001

SC-002			
Fuente:	UR-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	El usuario se autenticará con email y contraseña.		

Tabla 43. SC-002

SC-003			
Fuente:	UC-003, UR-001		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Cuando se introduzca una nueva búsqueda en el motor de búsqueda Google, se comprobará en la base de datos si existe información almacenada sobre búsquedas previas con idénticas palabras clave.		

Tabla 44. SC-003

3.3.3.6. Requisitos de aceptación de las pruebas

Especifican las condiciones para la validación del software, es decir, cómo se debe verificar que el software cumple con los requisitos establecidos.

SA-001			
Fuente:	UR-003, UC-002, UC-003		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Cuando un usuario no tiene sesión iniciada no se mostrará la búsqueda alternativa ofrecida por la herramienta.		

Tabla 45. SA-001

SA-002			
Fuente:	UC-004		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Un sitio web únicamente será añadido al camino cuando su página previa pertenezca al camino.		

Tabla 46. SA-002

3.3.3.7. Requisitos de documentación

Especifican los requisitos para la documentación del proyecto.

SD-001			
Fuente:	UR-004		
Necesidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Se realizará un manual de usuario detallando cómo agregar la herramienta al navegador y cómo el usuario puede hacer uso de ella.		

Tabla 47. SD-001

3.3.3.8. Requisitos de seguridad

Especifican los requisitos para prevenir el sistema contra amenazas de confidencialidad, integridad y disponibilidad.

SS-001			
Fuente:	UR-005		
Necesidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	Durante toda la vida del proyecto		
Descripción:	Los datos de los usuarios estarán protegidos como refleja la LOPD.		

Tabla 48. SS-001

3.3.4. Matriz de trazabilidad

La matriz de trazabilidad nos permite comprobar que todos los requisitos de usuario han sido contemplados en los requisitos de software. De igual forma nos permitirá observar cómo todos los requisitos de software son extraídos a partir de los requisitos de usuario.

	UC-001	UC-002	UC-003	UC-004	UC-005	UR-001	UR-002	UR-003	UR-004	UR-005
SF-001	x									
SF-002							x			
SF-003								x		
SF-004		x	x			x				
SF-005			x							
SF-006			x	x						
SF-007		x	x			x				
SF-008				x		x				
SF-009				x						
SF-010				x						
SF-011					x					
SI-001							x			
SI-002								x		
SI-003								x		
SI-004								x		
SI-005							x			
SI-006		x				x				
SO-001							x			
SO-002								x		
SO-003			x			x		x		
SO-004			x	x						
SO-005										
SR-001	x									
SR-002				x			x			
SC-001							x			
SC-002								x		
SC-003			x			x				
SA-001		x	x					x		
SA-002				x						
SD-001									x	
SS-001										x

Tabla 49. Matriz de trazabilidad. Requisitos usuario / Requisitos software

4. DISEÑO E IMPLEMENTACIÓN

En este capítulo es definido el proceso de desarrollo llevado a cabo para la construcción de la herramienta. El diseño, primera fase del proceso, proporcionará una idea completa del software, sirviendo como guía para la etapa de codificación o implementación del sistema.

4.1. Diseño

La finalidad del proceso de diseño es la descripción de la arquitectura del sistema y de la tecnología empleada para su desarrollo.

4.1.1. Arquitectura del sistema

Para el desarrollo de la herramienta ha sido elegido un modelo de arquitectura conocida como arquitectura multinivel. Este estilo de programación está centrado en el objetivo de separar la lógica de diseño de la lógica de negocio.

Su principal ventaja consiste en que el desarrollo del sistema se puede efectuar en varios niveles. Consiguiendo de esta forma que en caso de que se deba realizar alguna modificación, únicamente se tenga que alterar el nivel requerido sin la necesidad de revisar entre código mezclado.

Actualmente el diseño más utilizado es el basado en tres niveles o capas y será el que utilizemos para el desarrollo de nuestro sistema. Seguidamente mostramos una ilustración sobre esta arquitectura:

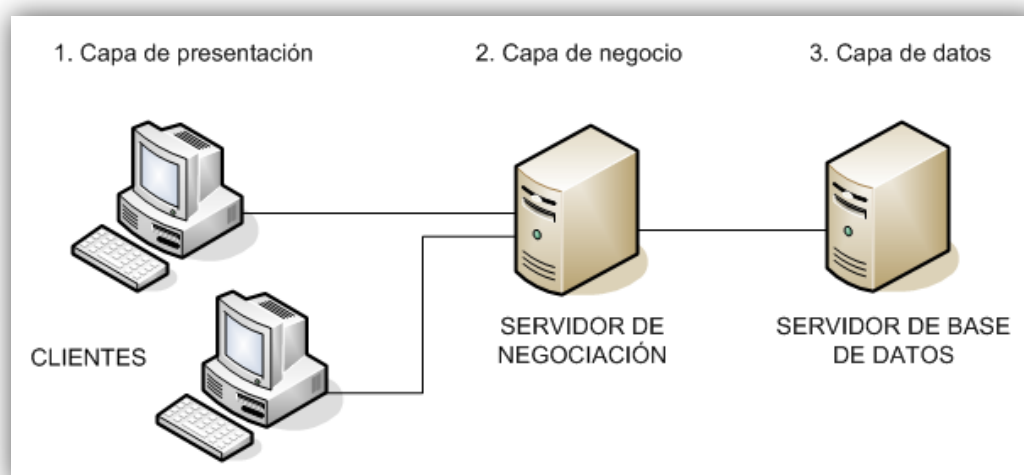


Ilustración 7. Arquitectura del sistema

Cada una de las capas tendrá la siguiente función en nuestro sistema:

- *Capa de presentación.* Capa que se muestra al usuario, le comunica la información y captura la información del usuario en un mínimo proceso. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa de negocio.

Es el código de la *Extensión de Chrome*, los archivos que el usuario descargará e instalará en su navegador. Será la encargada de capturar una búsqueda del usuario en Google y enviar la petición a la capa de negocio para obtener los resultados ofrecidos por nuestra herramienta. A su vez será la encargada de enviar a la capa de negocio la información sobre las rutas seguidas por los usuarios en busca de sitios web de su interés.

- *Capa de negocio.* Capa donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso para que los resultados sean presentados. Se denomina lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse.

Estará alojada en un servidor web. Deberá proporcionar mecanismos que permitan el establecimiento de sesiones de forma que se puedan implementar transacciones y operaciones unitarias. Recibirá las peticiones que realice la capa de presentación y realizará las solicitudes necesarias a la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

- *Capa de datos.* Aquí es donde residen los datos y es la encargada de acceder a los mismos. Está formada por un gestor de bases de datos que realiza todo el almacenamiento de datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

4.1.2. Tecnologías empleadas

Para el desarrollo de la *Extensión de Chrome* nos basaremos en el tutorial oficial de Google sobre los primeros pasos en la construcción de una extensión [23].

Inicialmente este tutorial nos recomienda estar familiarizados con tres tecnologías, *HTML*, *CSS* y *JavaScript*, que serán la base para el desarrollo de la extensión. En nuestra implementación también será necesaria la tecnología *jQuery* para la captura de eventos producidos durante la navegación y *AJAX* para la comunicación de la capa de negocio.

En el servidor necesitaremos la utilización del lenguaje *PHP* para la recepción y respuesta de las peticiones por parte de la capa de presentación y la realización de acciones sobre la capa de datos, donde para el almacenamiento de estos se utilizará una base de datos *MySQL*.

4.1.2.1. HTML

HTML (*HyperText Markup Language*) es un sistema que se utiliza para la construcción de las páginas web. Es una serie de *tags* o etiquetas que sirven para definir la estructura de los documentos.

No es un lenguaje de programación puesto que no realiza ningún tipo de operación o función y no es procesado por compiladores, sino que es un lenguaje de marcado cuya finalidad es que un navegador sea capaz de leer el contenido para poder interpretarlo y mostrarlo en la pantalla de los usuarios, por lo que es importante que el documento esté bien formado.

La estructura que siguen prácticamente todos los documentos HTML es la siguiente:

```
<HTML>
  <HEAD>
    <TITLE>Titulo Página</TITLE>
  </HEAD>
  <BODY>
    Cuerpo del documento
  </BODY>
</HTML>
```

Ilustración 8. Estructura general de un documento HTML

Las etiquetas que se pueden apreciar en la imagen son las que definen la estructura general de los documentos. Por cada una de estas etiquetas los navegadores interpretan el siguiente significado:

- `<HTML>`, se sitúan al principio y al final del documento delimitando dónde se encuentra el contenido.
- `<HEAD>`, abarca la cabecera del documento. La cabecera puede contener, además del título de la página, enlaces a archivos externos como pueden ser las hojas de estilo y etiquetas `<meta>` que sirven para indicar el contenido de la web y que son utilizadas por los motores de búsqueda.
- `<TITLE>`, indica el título de la página, el cual será descriptivo preferiblemente.
- `<BODY>`, engloba el cuerpo principal del documento, el contenido del mismo.

En particular, para la implementación de nuestra herramienta se ha utilizado HTML5, el cual con una de sus nuevas capacidades, nos permite el almacenamiento local de los datos en el lado del cliente.

4.1.2.2. CSS

CSS (*Cascading Style Sheets*) es un lenguaje de estilo utilizado para definir la apariencia de los documentos HTML. CSS se utiliza para formatear el contenido de los documentos previamente estructurados. Fue desarrollado para diferenciar la presentación de la estructuración de las páginas HTML.

CSS es la forma de indicar las fuentes, tamaño y colores de los distintos textos contenidos en el documento, los márgenes, las imágenes o colores del fondo de la página, la anchura y altura tanto de la página como de las cajas contenidas en la misma y otros muchos factores.

4.1.2.3. JavaScript

JavaScript es un lenguaje que funciona del lado del cliente y los navegadores serán los intérpretes de estos códigos. Fue diseñado para añadir interactividad con los usuarios en las páginas web debido a que HTML únicamente permitía crear páginas estáticas.

Algunas de sus principales características son:

- Permite la declaración de variables, condiciones y bucles.
- Los programadores pueden desarrollar sus propias funciones.
- Admite la programación orientada a objetos.

Este lenguaje permite realizar distintas acciones en el lado del cliente como por ejemplo la comprobación de los distintos campos de un formulario o mostrar mensajes de alerta al usuario, consiguiendo así la interacción con el usuario.

La inclusión de JavaScript en las páginas HTML se realiza con la utilización de la etiqueta `<script type="text/javascript">` de forma que cuando el navegador lo lea interpretará lo contenido entre estas etiquetas como lenguaje JavaScript.

4.1.2.4. jQuery

jQuery son unas librerías de JavaScript que contienen funciones que permiten simplificar la programación en JavaScript para interactuar con las páginas HTML.

Estas librerías permiten la captura de eventos producidos sobre el documento HTML, la manipulación del DOM y agrega la posibilidad de interactuar con AJAX.

4.1.2.5. DOM

El DOM (*Document Object Model*) es la estructura de objetos que se genera cuando el navegador carga un documento HTML. Define la estructura lógica de la página web y el modo en cómo se accede y manipula el documento.

El DOM permite a los programadores navegar por la estructura del documento y consultar, modificar, añadir o eliminar los objetos y contenidos de éste.

4.1.2.6. AJAX

AJAX (*Asynchronous JavaScript And XML*) es una técnica para el desarrollo web, diseñada para la creación de aplicaciones interactivas.

Para aumentar la interactividad, usabilidad y velocidad de una página web, permite realizar cambios sobre la misma sin la necesidad de recargar la página, gracias a la comunicación asíncrona de los documentos HTML con el servidor en un segundo plano.

4.1.2.7. JSON

JSON (*JavaScript Object Notation*) es un formato ligero de intercambio de información, independiente del lenguaje de programación [24]. Está construido por dos estructuras:

- Una colección de pares de nombre/valor. En otros lenguajes esto sería conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Almacena datos de forma organizada, con estructuras universales que todos los lenguajes de programación soportan y que en JSON se representan con las siguientes formas:

- *Objeto*. Conjunto desordenado de pares nombre/valor. Un objeto comienza con “{” -llave de apertura- y termina con “}” -llave de cierre-. Cada nombre es seguido por “:” -dos puntos- y los pares nombre/valor están separados por “,” -coma-.
- *Array*. Colección de valores. Un *array* comienza con “[” -corchete izquierdo- y termina con “]” -corchete derecho-. Los valores se separan por “,” -coma-.

- *Valor*. Puede ser una cadena de caracteres con acotada por “” -comillas dobles-, un número, un booleano, `null`, un objeto o un *array*. Estas estructuras pueden anidarse.

4.1.2.8. PHP

PHP (*Hipertext Preprocessor*) es un lenguaje interpretado que se ejecuta en el lado del servidor, que nos permite acceder a los recursos contenidos en el servidor como por ejemplo una base de datos.

El código PHP es utilizado para la generación dinámica de páginas web y suele incluirse en el contenido de un documento HTML. El cliente únicamente recibirá una página HTML resultado de la ejecución en el lado del servidor de la página PHP.

4.1.2.9. MySQL

MySQL es un gestor de bases de datos muy popular en aplicaciones web. Emplea el lenguaje SQL para la realización de sus consultas a la base de datos, permitiendo la interacción con lenguajes de programación como PHP.

4.1.2.10. Tareas programadas

Las Tareas programadas ejecutan servicios en segundo plano. Permiten planificar distintas tareas para ser ejecutadas en una fecha determinada o en intervalos regulares de tiempo.

4.1.3. Entorno de desarrollo

Para la realización del Trabajo Fin de Grado se ha utilizado un equipo con las siguientes características:

- Procesador: Intel Core i7 3615QM 2,3 GHz
- Memoria RAM: 4GB
- Disco duro libre: 320GB

Seguidamente exponemos el software necesario para la implementación de la herramienta y la redacción de la documentación:

- Sistema Operativo: Windows 7
- Navegador: Google Chrome
- Editor: Notepad++
- Servidor web local: Apache 2.4.3

- Base de datos: MySQL 5.5.27
- Administrador MySQL: phpMyAdmin 3.5.2.2
- PHP 5.4.7
- HTML5
- Microsoft Office 2010
- Gestor de proyectos: SmartSheet

Durante su desarrollo será necesaria la instalación de la herramienta en Google Chrome para la realización de pruebas. En este sentido debemos ir a la página de administración de extensiones⁷, seleccionar que se desea trabajar en “Modo de desarrollador” e indicar la ruta de la carpeta donde se encuentra nuestra extensión con todos sus archivos tras hacer clic en el botón “Cargar extensión descomprimida...”.

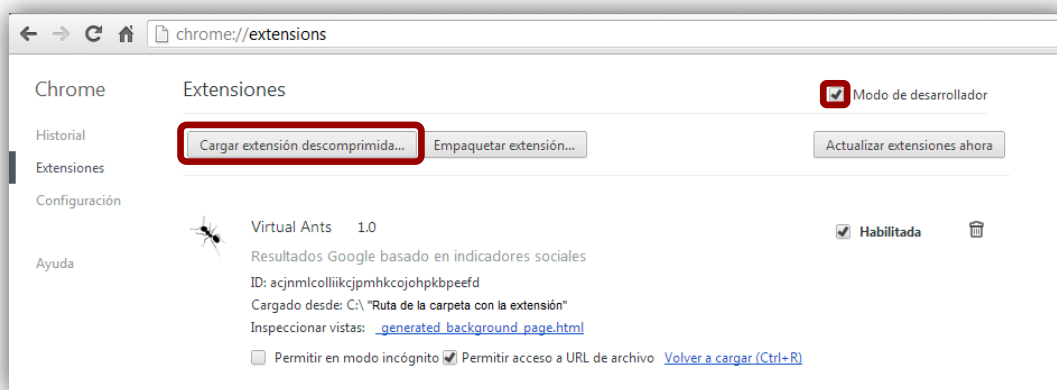


Ilustración 9. Página de administración de extensiones (Modo desarrollador)

⁷ <chrome://extensions/>

4.1.4. Entorno de implantación

Para poner a disposición de los usuarios la herramienta, en primer lugar será necesario empaquetar la extensión. Para ello nos debemos dirigir a la página de administración de extensiones, en “Modo de desarrollador” y en esta ocasión hacer clic sobre empaquetar la extensión. Aparecerá la siguiente ventana:

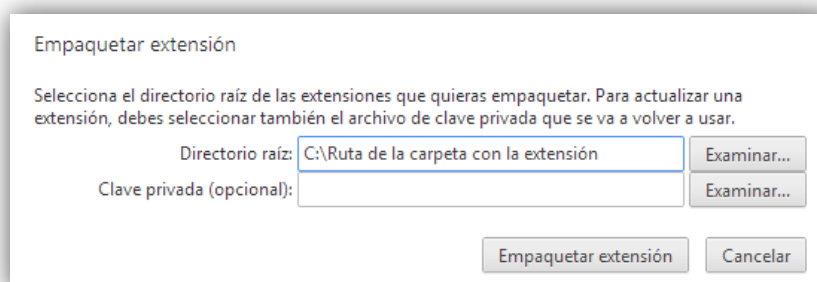


Ilustración 10. Ventana para Empaquetar extensión

En “Directorio raíz” indicamos la ruta de la carpeta y seleccionamos “Empaquetar extensión”. Chrome nos crea dos archivos:

- “.crx”, archivo de la extensión.
- “.pem”, archivo que servirá para las futuras actualizaciones de la extensión.

El archivo “.crx” será el proporcionado a los usuarios para la instalación de la herramienta. A su vez los usuarios deberán cumplir las *Condiciones de entorno* anteriormente descritas.

Para el alojamiento de las lógicas de negocio y de datos se utilizará un servidor con las siguientes características:

- Procesador: 2 Intel Quad-Core Xeon 2.13 Ghz
- Memoria RAM: 8 Gb ECC
- Disco duro: 2 HD SATA-II 1.000 Gb Hot plug
- Sistema Operativo: Windows 2008

4.2. Implementación

Como hemos visto en la descripción de la *Arquitectura del sistema* la herramienta estará dividida en tres niveles. Cada uno de éstos niveles se desarrollará de manera independiente.

4.2.1. Extensión de Chrome

En la capa de presentación implementaremos una extensión de Chrome que permita a los usuarios de la herramienta interactuar con las páginas web.

Para empezar su desarrollo Google Chrome nos pide detallar un archivo en formato JSON nombrado como `manifest.json`. En este archivo declaramos las variables de configuración de nuestra extensión, reflejando las posibles acciones que podrá realizar la herramienta.

A continuación detallamos cuales son los campos contenidos por nuestro fichero de configuración:

- **name.** Campo obligatorio compuesto por una cadena de texto, no superior a 45 caracteres, que identifique la extensión. Nuestra extensión tendrá el nombre de “Virtual Ants”.
- **description.** Campo recomendado compuesto por una cadena de texto, no superior a 132 caracteres, que debe proporcionar una descripción breve de la extensión.
- **version.** Campo obligatorio que identifica la versión de la extensión. El sistema de actualización automática de Google Chrome compara versiones para determinar si una extensión instalada debe ser actualizada. Inicialmente nuestra herramienta tendrá una versión “1.0”.
- **manifest_version.** Campo obligatorio que especifica la versión del formato del archivo. Actualmente todas las extensiones que sean desarrolladas deben contener “`manifest_version`”: 2”.
- **browser_action.** Campo opcional utilizado para situar el icono en la barra de herramientas del navegador. Se puede añadir la posibilidad de abrir una ventana emergente cuando el usuario hace clic sobre el icono. Usaremos este campo para proporcionar la funcionalidad de registro y autenticación de usuarios a la aplicación.

- **background.** Campo para agregar páginas de fondo, partes de la extensión que se ejecutan en un segundo plano de manera constante.
- **permissions.** Campo para la declaración de permisos necesarios para la utilización de APIs de Chrome en la extensión. Nuestra aplicación necesitará los siguientes permisos:
 - "<all_urls>", que nos permite trabajar sobre todos los sitios web.
 - "tabs", que nos permite trabajar sobre las pestañas del navegador.
 - "bookmarks", que nos permite trabajar sobre los marcadores (favoritos) de los usuarios.
 - "storage", que nos permite el almacenamiento de datos de forma local, en el lado del cliente.
- **content_scripts.** Campo para definir archivos JavaScript que se ejecutan en el contexto de páginas web. Utiliza el estándar DOM lo que permite leer los objetos de las páginas web visitadas e incluso hacer cambios en los mismos.
- **icons.** Campo recomendado representativo de la extensión. Se debe proporcionar el icono en distintos tamaños. A continuación mostramos la imagen utilizada como icono de nuestra herramienta:

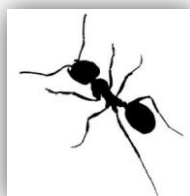


Ilustración 11. Icono de Virtual Ants

Una vez configurado el `manifest.json` creamos la página HTML `popup.html` que contendrá código JavaScript encargado de decidir que página mostrar al usuario. Seguidamente explicamos las posibles opciones:

- Cuando no haya sesión iniciada se mostrará la página de inicio de la herramienta (`index.html`), donde el usuario podrá iniciar sesión con su email y contraseña.

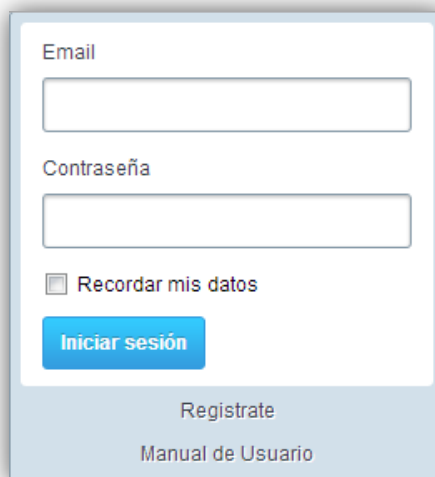


Ilustración 12. Página de Inicio (Virtual Ants)

Un fichero JavaScript (`sesión.js`) recogerá el valor de los campos rellenos al pulsar la tecla *Iniciar Sesión* y los enviará mediante AJAX a la lógica de negocio.

- En caso de que el usuario tenga su sesión iniciada, mostraremos una ventana al usuario donde se le dará la bienvenida a la herramienta (`logged.html`).

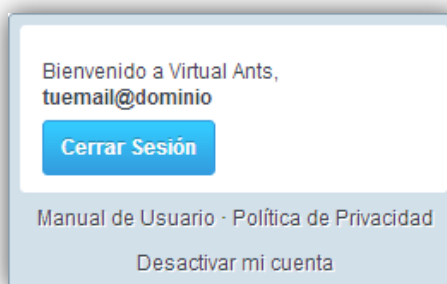


Ilustración 13. Página de Bienvenida (Virtual Ants)

- En caso de que los datos introducidos en el inicio de sesión sean incorrectos se mostrará un mensaje al usuario indicándole la causa del error en el inicio de sesión.

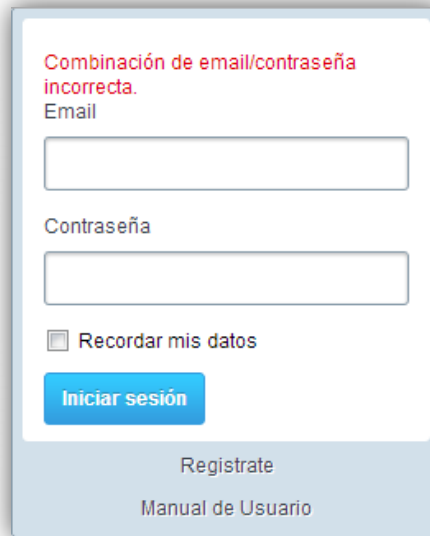


Ilustración 14. Autenticación incorrecta (Virtual Ants)

- Cuando se le muestre la pantalla de inicio a un usuario no registrado, éste podrá acceder a una página (`register.html`) para completar su registro en la herramienta.



Ilustración 15. Página de Registro (Virtual Ants)

Al hacer clic sobre el botón *Registrarse* un archivo JavaScript (`register.js`) comprobará que ambas contraseñas coinciden y que el usuario acepta la *Política de Privacidad* para realizar la petición de registro a la capa de negocio.

En todo momento el usuario tendrá la posibilidad de dar de baja su cuenta de la aplicación. Para ello una vez tiene sesión iniciada dispondrá de la opción para desactivar su cuenta.

Para el diseño de la interfaz gráfica se ha creado el fichero `style.css` con el cual logramos definir la apariencia de las páginas HTML de la extensión.

Por último, la extensión tendrá una página de fondo `background.js` en la cual estará contenida su funcionalidad:

- Cuando el usuario ingrese en un sitio web comprobará si la página accedida es el motor de búsqueda Google.
- Cuando el usuario se encuentra en Google y ha realizado una consulta, se enviará una petición AJAX al servidor con las *keywords* o palabras clave introducidas por el usuario. Si la herramienta contiene datos de búsquedas realizadas por usuarios anteriores con las mismas palabras clave se modificará el DOM de la página de resultados de Google y se mostrará un ranking alternativo de enlaces.

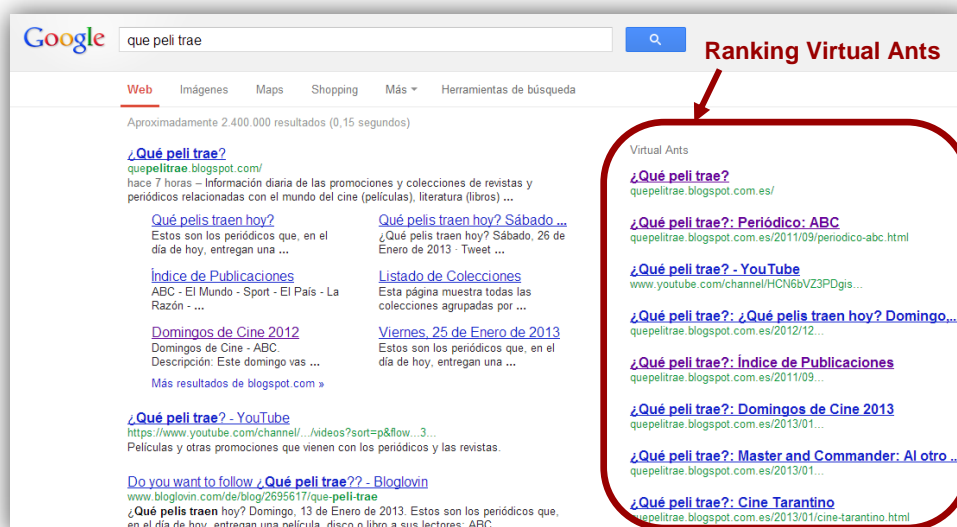


Ilustración 16. Página de Resultados de Google (Virtual Ants)

En caso de que no exista la búsqueda realizada en nuestra base de datos no se producirá ningún cambio en la página de resultados.

- Tras una consulta del usuario en Google el usuario comienza a visitar distintos sitios web a través de los enlaces ofrecidos en busca del sitio que le ofrezca la información de su interés. Al ingresar en una página se recuperará el recorrido almacenado en la base de datos para la consulta realizada, creado a partir de las consultas realizadas por usuarios anteriores. Se comprobará si el sitio web visitado se encuentra en ese recorrido, en cuyo caso aumentará el valor o “feromona” de esa página en el recorrido de la consulta.
- Cuando la web visitada no se encuentra en dicho recorrido, la herramienta obtendrá su página previa, es decir, la página anterior mediante la que se ha llegado al sitio actual, dando lugar a la realización de distintas operaciones:
 - ❖ En caso que la web actual no tenga página anterior, no se realizan más operaciones sobre la página, puesto que se ha llegado a ella sin la navegación a través de enlaces, siendo imposible llegar a ella tras una consulta en Google.
 - ❖ Cuando existe un sitio web previo se comprobará si es Google, lo que significaría que el usuario ha llegado a la web actual a través de hacer clic en un enlace ofrecido por una búsqueda realizada, por lo que el sitio web pasaría a formar parte del recorrido.
 - ❖ Si la página anterior es distinta de Google, se verificará si esa página previa se encuentra en el recorrido existente, en cuyo caso añadiríamos la página actual al recorrido, puesto que sería un sitio web al que se ha llegado navegando a través de distintos enlaces y cuyo origen es una consulta en Google.
 - ❖ Cuando su página previa no forma parte de la solución, la página será descartada.
- Además la página de fondo de la herramienta será la encargada de capturar los distintos eventos de los usuarios indicativos de que la página les ha resultado interesante, que provocarán un incremento en las “feromonas” depositadas en una página como por ejemplo pasar un tiempo determinado en un sitio web, añadir la página como *favoritos*⁸, recomendar la página en redes sociales o imprimir la página.

⁸ URL interesante para el usuario que almacena en su navegador para poder acceder directamente a ella en un futuro de una forma rápida y sencilla.

4.2.2. Lógica de negocio

Estará alojada en un servidor web, codificada en PHP y será la encargada de recibir las solicitudes realizadas por la extensión y devolver una respuesta. Además será la capa encomendada de realizar las distintas operaciones que se deban llevar a cabo con la base de datos.

Para empezar se creará un archivo que contiene la definición de todos los datos de conexión con base de datos (`connect_mysql.php`) y el cual incluiremos en el resto de ficheros PHP que necesiten conectarse. Estos datos de conexión son el nombre de la base de datos, la dirección del servidor en el cual está alojada, el nombre de usuario y la contraseña con la cual conectarse a la misma.

Existirán ficheros encargados del registro y autenticación de usuarios:

- Para el registro se ha creado un fichero (`register.php`) que obtendrá el email y contraseña, la cual es cifrada, introducidos por el usuario y realizará la petición de inserción en base de datos.

Responderá a la extensión si la inserción se ha realizado correctamente o si por el contrario ha existido algún error, de forma que se pueda informar al usuario sobre el mismo, como sucede cuando el email ya está registrado en la herramienta.

- Para el inicio de sesión se realizará una consulta a través del fichero `validateUser.php` comprobando si existe la combinación de email y contraseña en la base de datos. Si el usuario está registrado y la contraseña introducida es correcta se creará una sesión PHP para mantener la sesión del usuario activa. En caso de que la combinación de datos introducidos no se encuentre en la base de datos se responderá a la extensión para que informe al usuario del error.

El resto de ficheros PHP serán los encargados de la comunicación con la página de fondo de la extensión. Seguidamente describimos cuales son las funcionalidades de estos ficheros:

- Ha sido creado un fichero (`searchKeywords.php`) para recibir las *keywords* introducidas en Google y consultar en la base de datos si existe un recorrido creado por otros usuarios. En caso de que exista, se enviará dicho recorrido para su tratamiento en la extensión.

- Cuando nos encontramos en otra página distinta a Google se envía a la extensión el recorrido de la última consulta realizada para su tratamiento por parte de la página de fondo.
- Si la extensión concluye añadir la página web visitada al recorrido o actualizar los pesos en caso de que la página exista en dicho recorrido, se ha creado un archivo (`savePath.php`) para la actualización de los datos pertenecientes a éste recorrido.

Por último en el servidor se programará una tarea encargada de la evaporación de feromonas. Se creará un fichero que se ejecutará automáticamente en el servidor semanalmente con la siguiente funcionalidad:

- Recorrer las rutas de todas las consultas almacenadas en la base de datos comparando si la fecha actual y la fecha de última visita a una web o acción realizada sobre ella es superior a un tiempo establecido.
- Cuando el tiempo transcurrido entre las fechas es mayor que el especificado se producirá una disminución del peso asociado a una URL en el recorrido.

Para la evaporación de la feromona se eliminará un porcentaje de su valor actual. Con esta decisión pretendemos que sitios web que adquirieron un gran peso en un momento determinado, pero que en el momento actual no sean una buena recomendación, no perduren en los recorridos creados para una consulta.

4.2.3. Base de datos

En la capa de datos han sido creadas dos tablas:

- *Users*: una tabla que contendrá email y contraseña de los usuarios registrados.
- *Searches*: una tabla que contendrá las búsquedas realizadas por los usuarios en Google y el recorrido seguido por las “hormigas virtuales” para esas búsquedas.

En la tabla de usuarios tendremos el email como *Primary Key* (clave primaria) de forma que un mismo email no pueda ser registrado en más de una ocasión. En caso de que se produzca esta acción el gestor enviará el mensaje de error provocado a la lógica de negocio.

Por su parte, en la tabla de búsquedas, se utilizará la consulta de los usuarios como *Primary Key*. El valor de la clave primaria no puede ser repetido debido a que por cada búsqueda solo queremos tener un recorrido. De ésta forma cuando un usuario realiza una búsqueda con unas *keywords* ya existentes y comienza a visitar diferentes páginas web, en nuestra base de datos únicamente deberá actualizarse el recorrido.

En caso de que la consulta sea realizada por primera vez por un usuario de la herramienta, añadirá una nueva entrada en la tabla de búsquedas.

Las rutas seguidas por los usuarios en las búsquedas serán almacenadas en formato JSON, creando una matriz o *array* por cada búsqueda con las páginas visitadas que pertenezcan al recorrido seguido por los usuarios. Cada página perteneciente al recorrido será almacenada en el *array* con la URL, el título, el peso (“feromona”) y la fecha de la última visita o acción realizada en la página web.

5. MARCO REGULADOR

En el año 2007 la Agencia Española de Protección de Datos (AEPD) publicó un documento analizando las políticas de privacidad de los principales motores de búsqueda [25]. Nos basaremos en el mismo para conocer el conjunto de leyes y normas a tener en cuenta para la gestión de nuestra herramienta.

En este documento se cita que los buscadores “*son servicios de la sociedad de la información sujetos no sólo a las garantías de la Ley Orgánica 15/1999 de Protección de Datos (LOPD), sino también de la Ley 34/2002 de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSI), que engloba en este concepto ‘la provisión de instrumentos de búsqueda o de enlaces a otros sitios de Internet’*”.

Como conocemos la principal función de los buscadores es ofrecer unos resultados a los usuarios que realizan una consulta, pero además suelen ofrecer servicios personalizados para lo que el usuario se debe dar de alta proporcionando una dirección de correo electrónico y una contraseña.

Estos servicios de búsqueda fundamentan la obtención, uso y conservación de datos personales de los usuarios en base a las siguientes razones:

- *Mejorar el servicio de búsqueda a la vista de la experiencia de los usuarios.* El éxito de los buscadores recae en la capacidad para mostrar los resultados que puedan resultar más relevantes para el usuario, para lo que se pueden centrar en los hábitos realizados por los usuarios.
- *Garantizar la seguridad del servicio.* Los motores de búsqueda se exponen ataques informáticos que pueden afectar tanto al prestador del servicio como a los usuarios, como pueden ser los ataques de denegación de servicios (DoS).
- *Detección de otras actividades fraudulentas.* Prevenir que terceras personas pretendan alterar el orden de los resultados de las búsquedas en busca de un beneficio.
- *Conservación de la información a disposición de las autoridades competentes.* Para atender posibles requerimientos por las autoridades con el fin de investigar actividades ilegales.

Ante estas alegaciones la AEPD dispuso que el tratamiento de los datos deba cumplir con el sistema de garantías que establecen la LOPD y la LSSI en beneficio de los ciudadanos y de los usuarios de los servicios.

Cuando los usuarios del servicio deciden registrarse de manera voluntaria, la legitimación para el tratamiento de sus datos estaría fundamentada en el consentimiento de los usuarios. Aun así, el tratamiento de los datos debe estar sujeto a las finalidades que lo justifica como son la mejora del servicio, la seguridad y la detección de fraudes.

Por el contrario, no se puede justificar la conservación de los datos durante amplios periodos de tiempo por si en el futuro éstos fueran requeridos por autoridades competentes. No existe la posibilidad de almacenar los datos más allá del tiempo necesario para la prestación del servicio solicitado por el usuario o para el cumplimiento de la normativa vigente.

En España, esta normativa sobre la conservación de los datos se encuentra basada en la *Ley 25/2007, de 18 de octubre, de conservación de datos relativos a las comunicaciones electrónicas y a las redes públicas de comunicación* y únicamente con fines de investigación, sin que los motores de búsquedas se encuentren entre los servicios afectados por esta regulación.

Para hacer uso de nuestra herramienta los usuarios deberán realizar un registro previo, donde el único dato solicitado será un correo electrónico acompañado de una contraseña.

Según la definición de la LOPD [22] “*cualquier información concerniente a personas físicas identificadas o identificables*” es considerada como tal. El conjunto de signos que forman un correo electrónico permiten la asociación a una persona identificada o identificable y por lo tanto debe ser considerado como una información de carácter personal.

Para poder realizar un tratamiento con una dirección de email, deberemos disponer por lo tanto del consentimiento de los titulares de los datos. La utilización de este dato sin consentimiento vulneraría la normativa sobre protección de datos.

6. VALIDACIÓN DEL SISTEMA

En el presente capítulo serán especificadas las pruebas que deberán ser realizadas para la validación del sistema. Una vez que se han realizado pruebas parciales sobre los componentes individuales de la aplicación durante su desarrollo, probamos el sistema de forma global.

Seguidamente definimos el formato con el que mostraremos cada una de las pruebas identificadas, atendiendo al estándar IEEE 829-2008 [26]:

PR-Xnnn
Elementos de prueba:
Descripción:
Especificaciones de entrada:
Especificaciones de salida:
Requisitos especiales del procedimiento:

Tabla 50. Formato de especificación de prueba

Dentro del formato definido encontramos los siguientes atributos:

- *Identificador*: PR-Xnn. Donde:
 - PR indica que pertenece a un caso de prueba.
 - X define el tipo de prueba al que afecta.
 - nnn es el número identificativo de la prueba.
- *Elementos de prueba*: requisitos que son probados.
- *Descripción*: descripción de la prueba.
- *Especificaciones de entrada*: datos que se deben ingresar para la realización de la prueba.
- *Especificaciones de salida*: respuesta esperada por el sistema para que la prueba se considere como aceptada.
- *Requisitos especiales de procedimiento*: condiciones previas que deben existir para la realización de la prueba.

6.1. Pruebas funcionales

Este tipo de pruebas tienen como objetivo asegurar que la herramienta cumple todas las funciones que han sido dadas en la especificación de los *Requisitos de usuario*.

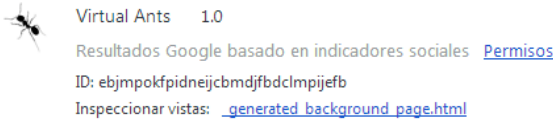
PR-F001	
Elementos de prueba:	UC-001
Descripción:	El usuario instalará correctamente la aplicación en el navegador.
Especificaciones de entrada:	Situarse en la página de administración de extensiones (chrome://extensions/). Arrastrar el archivo “.crx” hasta que aparezca el mensaje “Soltar para instalar”. Tras soltar aparecerá una ventana pidiendo la confirmación, hacemos clic sobre “Añadir”.
Especificaciones de salida:	<p>En la página de extensiones deberá haberse añadido:</p> 
Requisitos especiales del procedimiento:	Disponer del fichero “.crx” que contiene la extensión empaquetada.

Tabla 51. PR-F001

PR-F002	
Elementos de prueba:	UR-002
Descripción:	Los datos introducidos en el proceso de registro deben ser correctos, previamente a su almacenamiento en la base de datos.
Especificaciones de entrada:	<p>Pulsar sobre el icono de la herramienta existente en la barra de direcciones del navegador. En la ventana de inicio que es abierta se hace clic sobre “Regístrate”. Rellenar el formulario de registro de las siguientes formas:</p> <ol style="list-style-type: none"> 1. Con un <i>email</i> con un formato no válido. 2. Con una <i>contraseña</i> con un número de caracteres inferior a seis. 3. Con dos contraseñas válidas pero diferentes. <p>En cada uno de los casos presentados, pulsar sobre el botón “Registrarse” para completar el registro.</p>
Especificaciones de salida:	El sistema limpia los campos del formulario y muestra un mensaje de error indicando el campo que no ha sido rellenado correctamente impidiendo su registro en el sistema y almacenamiento en la base de datos.
Requisitos especiales del procedimiento:	La extensión debe estar instalada en el navegador Google Chrome. No tiene que existir ninguna sesión iniciada.

Tabla 52. PR-F002

PR-F003	
Elementos de prueba:	UR-002
Descripción:	El correo electrónico introducido en el formulario de registro de la herramienta no debe existir en la tabla de usuarios registrados en la base de datos.
Especificaciones de entrada:	Pulsar sobre el icono de la herramienta existente en la barra de direcciones del navegador. En la ventana de inicio que es abierta se hace clic sobre “Regístrate”. Rellenar el formulario de registro con el <i>email</i> “prueba@virtualants.es” y pulsar sobre el botón “Registrarse” para completar el registro.
Especificaciones de salida:	El sistema limpia los campos del formulario y muestra el siguiente mensaje para informar al usuario de que el registro no se ha realizado correctamente: “Este correo electrónico ya está registrado”.
Requisitos especiales del procedimiento:	Previamente se habrá introducido en la base de datos un usuario con el correo electrónico “prueba@virtualants.es”. La extensión debe estar instalada en el navegador Google Chrome. No tiene que existir ninguna sesión iniciada.

Tabla 53. PR-F003

PR-F004	
Elementos de prueba:	UR-003
Descripción:	La combinación email/contraseña introducida en la página de inicio debe coincidir con la existente en la base de datos.
Especificaciones de entrada:	<p>Pulsar sobre el icono de la herramienta existente en la barra de direcciones del navegador. En la página de inicio introducir los siguientes datos:</p> <ul style="list-style-type: none"> • Email: prueba@virtualants.es • Contraseña: incorrecta <p>Pulsar botón “Iniciar Sesión”</p>
Especificaciones de salida:	El sistema limpia los campos de la página de inicio y muestra el siguiente mensaje: “Combinación de email/contraseña incorrecta”.
Requisitos especiales del procedimiento:	Previamente se habrá introducido en la base de datos un usuario con el correo electrónico “prueba@virtualants.es” y la contraseña “aleatoria”. No tiene que existir ninguna sesión iniciada.

Tabla 54. PR-F004

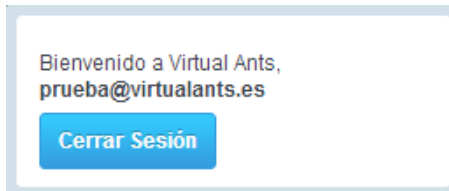
PR-F005	
Elementos de prueba:	UR-003
Descripción:	La herramienta debe mostrar una página de bienvenida cuando el usuario introduce la combinación email/contraseña correcta en el inicio de sesión.
Especificaciones de entrada:	<p>Pulsar sobre el icono de la herramienta existente en la barra de direcciones del navegador. En la página de inicio introducir los siguientes datos:</p> <ul style="list-style-type: none"> Email: prueba@virtualants.es Contraseña: aleatoria <p>Pulsar botón “Iniciar Sesión”</p>
Especificaciones de salida:	<p>El sistema muestra la siguiente página al usuario:</p> 
Requisitos especiales del procedimiento:	Previamente se habrá introducido en la base de datos un usuario con el correo electrónico “prueba@virtualants.es” y la contraseña “aleatoria”. No debe existir ninguna sesión iniciada.

Tabla 55. PR-F005

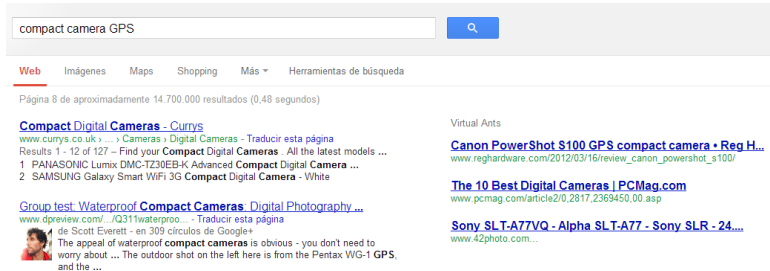
PR-F006	
Elementos de prueba:	UC-002, UC-003, UR-001
Descripción:	La herramienta modifica el DOM de la página de resultados de Google cuando se realiza una búsqueda.
Especificaciones de entrada:	Abrir https://www.google.es/ y realizar una búsqueda con los términos “compact camera GPS”.
Especificaciones de salida:	<p>Se debe mostrar la página de resultados de Google con un ranking alternativo al ofrecido por el motor de búsqueda con los enlaces recomendados por la herramienta.</p> 
Requisitos especiales del procedimiento:	Tener una sesión iniciada. La consulta realizada debe existir en nuestra base de datos con un recorrido realizado previamente.

Tabla 56. PR-F006

PR-F007	
Elementos de prueba:	UC-003, UR-001
Descripción:	La herramienta muestra un ranking de resultados alternativo ordenando los enlaces de forma descendente según su peso.
Especificaciones de entrada:	Abrir https://www.google.es/ y realizar una búsqueda con los términos “compact camera GPS”.
Especificaciones de salida:	Se debe mostrar la página de resultados de Google con un ranking alternativo al ofrecido por el motor de búsqueda con los enlaces recomendados por la herramienta ordenados de mayor a menor peso.
Requisitos especiales del procedimiento:	Tener almacenadas distintas URL para la búsqueda “compact camera GPS” con diferentes valores asignados a cada una de ellas. Se debe tener acceso a la base de datos para comprobar los pesos de las URL.

Tabla 57. PR-F007

PR-F008	
Elementos de prueba:	UC-004, UR-001
Descripción:	La herramienta asigna pesos a los sitios web que forman parte de la ruta que siguen los usuarios en busca de un sitio web de su interés.
Especificaciones de entrada:	Abrir https://www.google.es/ y realizar una búsqueda con los términos “compact camera GPS”. Visitar cualquiera de los enlaces ofrecidos por Google.
Especificaciones de salida:	En la tabla de búsquedas de la base de datos, para la búsqueda “compact camera GPS” debe añadirse el enlace visitado a la ruta con un peso asignado o en caso de que el sitio visitado ya formara parte de la solución habrá sido incrementado su peso.
Requisitos especiales del procedimiento:	Se debe tener acceso a la base de datos para comprobar los pesos de las URL.

Tabla 58. PR-F008

PR-F009	
Elementos de prueba:	UC-005
Descripción:	Los pesos asignados a las URL decrecerán cuando lleven un tiempo determinado sin ser visitados. Se comprobará si un sitio web con distancia entre fechas de última actualización y fecha actual superior al tiempo especificado disminuye su peso.
Especificaciones de entrada:	Ejecutar script en el servidor encargado de recorrer la tabla de búsquedas de la base de datos.
Especificaciones de salida:	El peso asignado a una URL con distancia entre fechas de última actualización y fecha actual superior al tiempo especificado deberá haber disminuido.
Requisitos especiales del procedimiento:	En la base de datos existirán registros de búsquedas almacenados.

Tabla 59. PR-F009

PR-F010	
Elementos de prueba:	UR-004
Descripción:	Los usuarios deberán poder acceder al “Manual de Usuario”.
Especificaciones de entrada:	Pulsar sobre el icono de la herramienta existente en la barra de direcciones del navegador. En la página de inicio pulsar sobre “Manual de Usuario”. Repetir procedimiento en la página de bienvenida.
Especificaciones de salida:	Se debe abrir una nueva pestaña en el navegador con el “Manual de usuario” en formato PDF.
Requisitos especiales de procedimiento:	--

Tabla 60. PR-F010

PR-F011	
Elementos de prueba:	UR-005
Descripción:	Un usuario únicamente podrá registrarse en la aplicación si acepta la Política de Protección de Datos de la herramienta.
Especificaciones de entrada:	Pulsar sobre el icono de la herramienta existente en la barra de direcciones del navegador. En la ventana de inicio que es abierta se hace clic sobre “Regístrate”. Rellenar el formulario de correctamente sin aceptar la <i>Política de privacidad</i> y pulsar sobre el botón “Registrarse” para completar el registro.
Especificaciones de salida:	Se informa al usuario al usuario de que el registro no se ha realizado correctamente y su motivo: “Debes aceptar la Política de privacidad y protección de datos”
Requisitos especiales de procedimiento:	No tener sesión iniciada.

Tabla 61. PR-F011

6.2. Pruebas de comunicación

Determinan que las interfaces entre los componentes del sistema funcionan adecuadamente.




PR-C001															
Elementos de prueba:	--														
Descripción:	Comunicación entre interfaz de usuario y lógica de negocio.														
Especificaciones de entrada:	Realizar una búsqueda aleatoria en https://www.google.es/														
Especificaciones de salida:	<p>En la herramienta de Chrome para desarrolladores se debe observar como se ha realizado una consulta a una página del servidor y ésta se ha completado correctamente.</p> <table><tr><th>Name</th><th>Method</th><th>Status</th></tr><tr><td>Path</td><td></td><td>Text</td></tr><tr><td> searchKeywords.php</td><td>POST</td><td>200</td></tr><tr><td>localhost</td><td></td><td>OK</td></tr></table>			Name	Method	Status	Path		Text	 searchKeywords.php	POST	200	localhost		OK
Name	Method	Status													
Path		Text													
 searchKeywords.php	POST	200													
localhost		OK													
Requisitos especiales de procedimiento:	Abrir la herramienta para desarrolladores de Chrome. Para ello utilizar el atajo “Ctrl+Shift+J” y situarse sobre la pestaña “Network”.														

Tabla 62. PR-C001

PR-C002	
Elementos de prueba:	--
Descripción:	Comunicación entre la lógica de negocio y la base de datos.
Especificaciones de entrada:	Realizar el registro de un nuevo usuario con los datos correctos. Los datos rellenados se envían a la lógica de negocio que los tratará y será la encargada de realizar la inserción en la base de datos.
Especificaciones de salida:	Observar en la tabla de usuarios de la base de datos como ha sido añadido un nuevo usuario con el email indicado al completar el formulario de registro.
Requisitos especiales de procedimiento:	Abrir la herramienta <i>phpMyAdmin</i> para visualizar los registros contenidos en la base de datos.

Tabla 63. PR-C002

6.3. Pruebas de aceptación

Se extraen de los *Requisitos de aceptación de las pruebas*, son definidas por el usuario del sistema y van encaminadas a comprobar que el sistema cumple con los requisitos de funcionamiento esperados.

PR-A001	
Elementos de prueba:	SA-001
Descripción:	El ranking de búsqueda alternativo únicamente será mostrado cuando exista una sesión iniciada.
Especificaciones de entrada:	Realizar una búsqueda aleatoria en https://www.google.es/
Especificaciones de salida:	Observar que en la página de resultados de Google no se muestran un ranking de enlaces alternativo a la búsqueda. En la herramienta de Chrome para desarrolladores debe aparecer el mensaje “[Virtual Ants] : Sesión no iniciada”
Requisitos especiales de procedimiento:	No tener ninguna sesión iniciada. Abrir la herramienta para desarrolladores de Chrome. Para ello utilizar el atajo “Ctrl+Shift+J” y situarse sobre la pestaña “Console”.

Tabla 64. PR-A001

PR-A002	
Elementos de prueba:	SA-002
Descripción:	Un sitio web únicamente pertenecerá al sendero construido por los usuarios a partir de una consulta cuando sea un enlace mostrado en la página de resultados de Google para esa búsqueda o su página anterior forme parte del sendero.
Especificaciones de entrada:	Realizar los siguientes pasos: 1. Realizar una búsqueda aleatoria en https://www.google.es/ 2. Abrir una nueva pestaña del navegador e introducir una dirección web que no se encuentre entre los enlaces ofrecidos en Google.
Especificaciones de salida:	Consultar el registro de la base de datos con las palabras clave de la búsqueda realizada en Google. En la ruta almacenada para este recorrido no debe aparecer la URL visitada en la nueva pestaña del navegador.
Requisitos especiales de procedimiento:	Tener una sesión iniciada. Abrir la herramienta <i>phpMyAdmin</i> para visualizar los registros contenidos en la base de datos.

Tabla 65. PR-A002

7. GESTIÓN DEL PROYECTO

Se expone en este capítulo la planificación seguida para el desarrollo de la herramienta y coste que ha supuesto su creación.

7.1. Planificación

En este punto se detallarán las principales fases planificadas para el desarrollo del Trabajo Fin de Grado, para las que se ha tenido como base la metodología de Métrica v3 [27]. Una metodología que nos ayuda en la planificación, desarrollo y mantenimiento de sistemas de información.

Es necesario realizar una planificación con los hitos que se pretende cumplir a lo largo del ciclo de vida del proyecto para prevenir desbordamientos de agendas y de costos a la vez que se consigue un resultado de calidad.

Inicialmente mostramos el ciclo de vida del proyecto donde se pueden observar las principales fases planificadas para el mismo una vez ha sido planteado el problema al que se pretende ofrecer una solución:



Ilustración 17. Ciclo de vida del Trabajo Fin de Grado

A lo largo del ciclo de vida del proyecto se irá realizando la documentación perteneciente a la memoria del Trabajo Fin de Grado donde se expondrán cada una de las fases por las cuales irá atravesando el proyecto.

7.1.1. Creación del Plan de Proyecto

En primer lugar es necesario definir tareas a efectuar, estimando el tiempo necesario para su realización, por lo que se necesita incorporar sus fechas de inicio y fin. Se detallarán hitos en los cuales se deberá presentar el desarrollo alcanzado en esa etapa para su evaluación.

Seguidamente mostramos una tabla con las fases definidas, en relación a las identificadas en el ciclo de vida del proyecto, y las tareas que las componen, junto a las fechas de inicio y fin planificadas para cada una y la duración en días de las mismas, la finalización de una fase constituirá la consecución de un hito.

Nombre de la tarea	Fecha de Inicio	Fecha de Fin	Duración (días)
Planteamiento Inicial	08/01/13	17/01/13	8
Planteamiento del problema	08/01/13	10/01/13	3
Objetivos	11/01/13	14/01/13	2
Gestión del Proyecto	15/01/13	17/01/13	3
Estado del Arte	18/01/13	07/02/13	15
Posicionamiento en buscadores	18/01/13	24/01/13	5
Algoritmos sociales	25/01/13	01/02/13	6
Investigación ACO	04/02/13	07/02/13	4
Análisis del Sistema	08/02/13	26/02/13	13
Propuesta	08/02/13	15/02/13	6
Especificación de requisitos	18/02/13	22/02/13	5
Marco regulador	25/02/13	26/02/13	2
Diseño del Sistema	27/02/13	07/03/13	7
Arquitectura del sistema	27/02/13	06/03/13	6
Entorno de implantación	07/03/13	07/03/13	1
Implementación del sistema	08/03/13	12/04/13	23
Codificación de la herramienta	08/03/13	10/04/13	21
Pruebas	01/04/13	12/04/13	10
Documentación	08/01/13	18/04/13	70
Memoria TFG	08/01/13	16/04/13	68
Manual de usuario	17/04/13	18/04/13	2

Tabla 66. Tareas identificadas en la planificación del proyecto

7.1.2. Diagrama de Gantt

Para visualizar gráficamente el tiempo planificado para el desarrollo de las distintas fases y sus tareas se ha utilizado un diagrama de Gantt. Utilizaremos esta herramienta para realizar estimaciones sobre el tiempo requerido por cada una de las tareas, así como del total del tiempo necesario para el completo desarrollo del trabajo.



La representación de los tiempos en el diagrama está indicada en días, por lo tanto deberíamos indicar que la media de horas dedicadas al día es de aproximadamente seis (6). Repartidas en cuatro (4) para la realización de la tarea concreta y dos (2) para la elaboración de la documentación, que como se ha indicado anteriormente se ha desarrollado paralelamente.

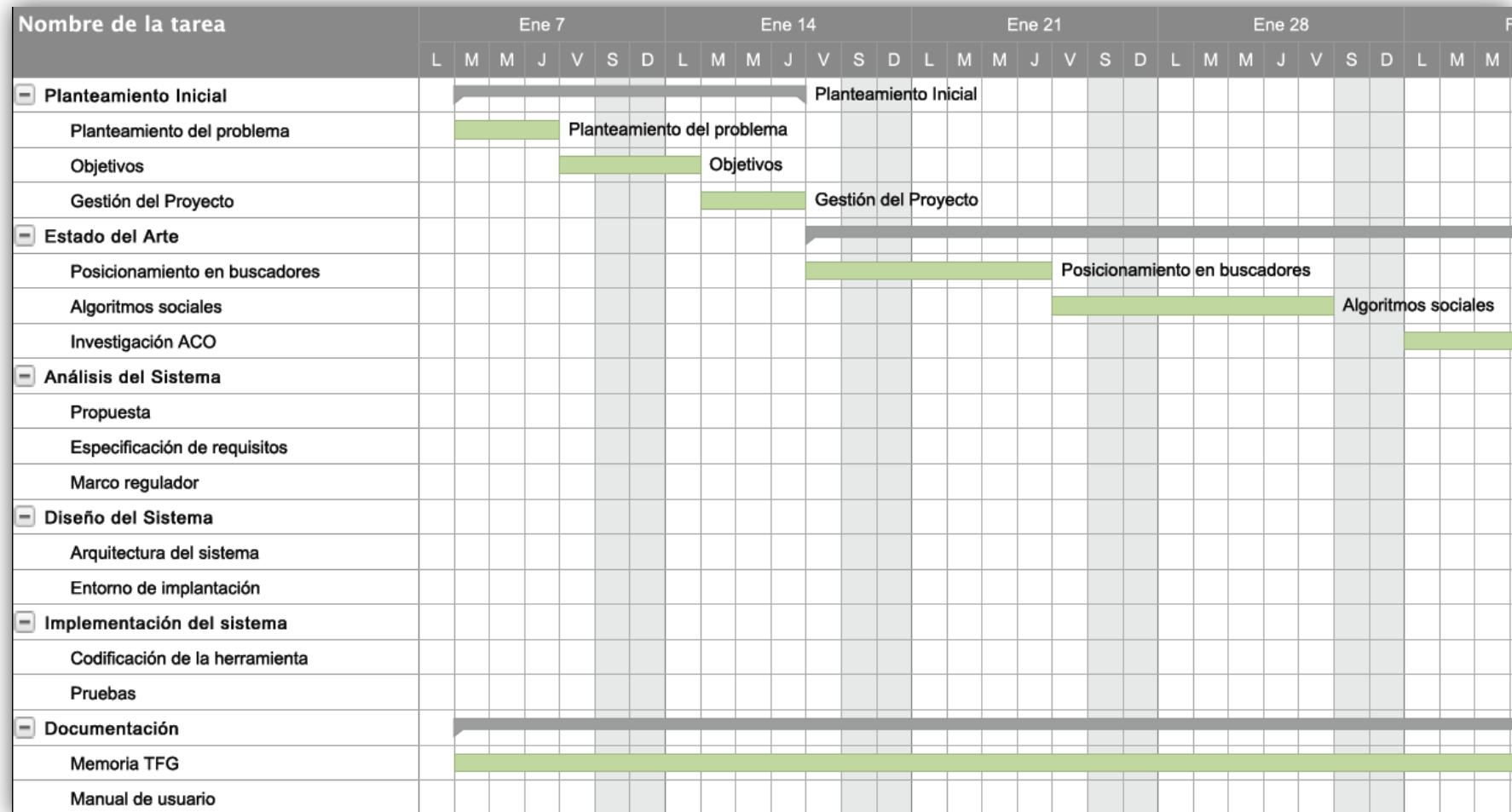


Ilustración 18. Diagrama de Gantt (1)

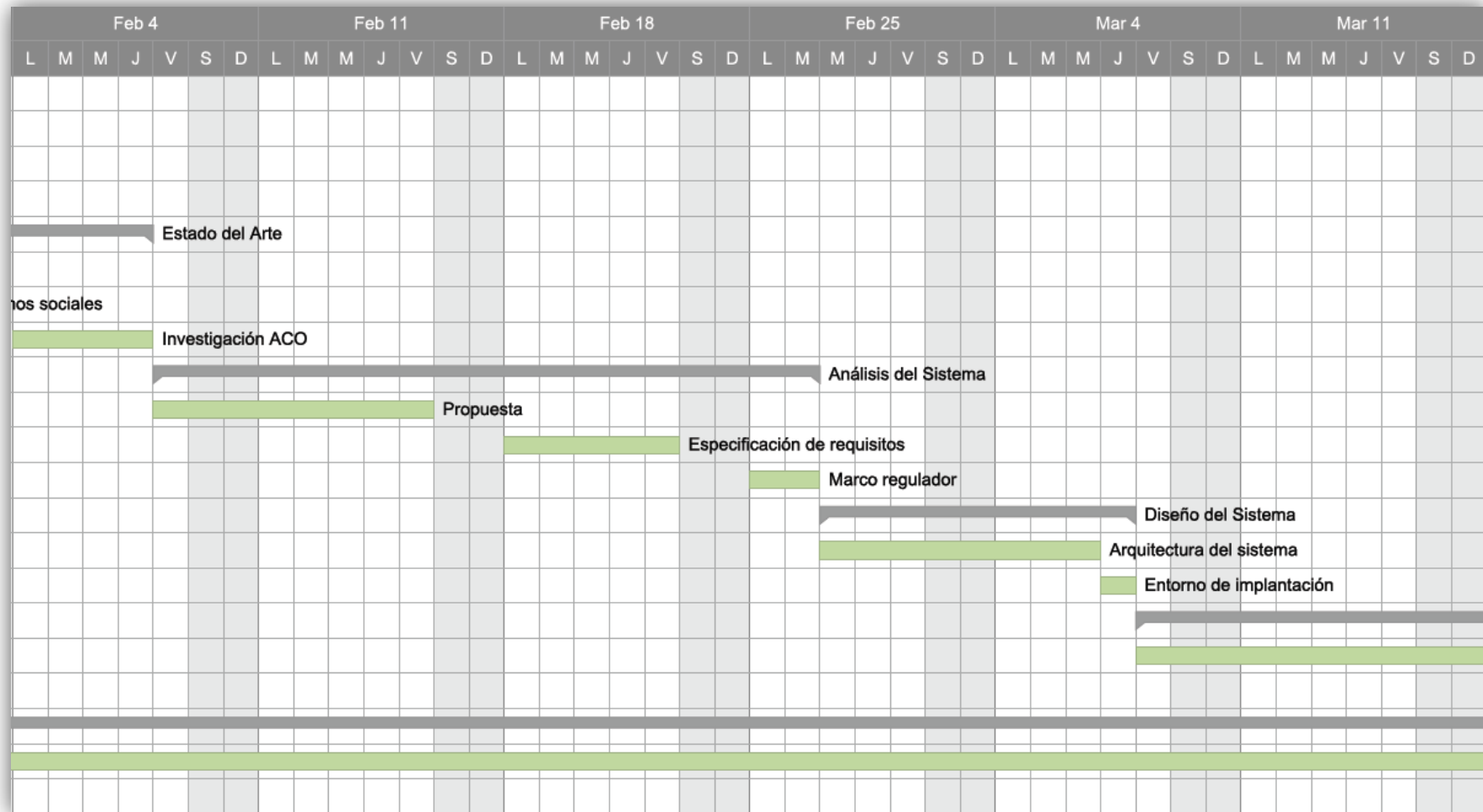


Ilustración 19. Diagrama de Gantt (2)

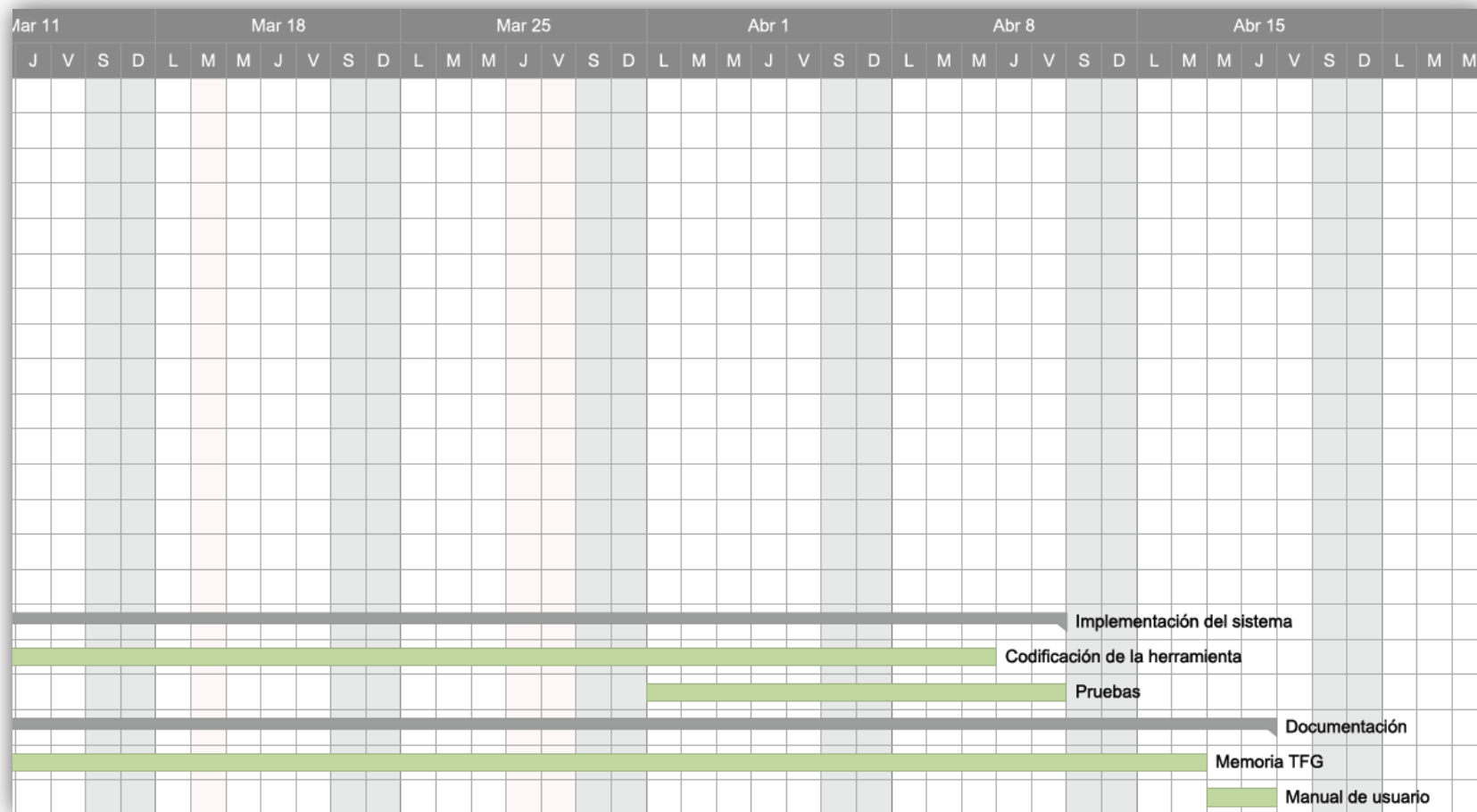


Ilustración 20. Diagrama de Gantt (3)

Se puede observar en el diagrama como durante 8 días coinciden pruebas con desarrollo de la herramienta. Esto se debe a que cada uno de esos días se dedicará un tiempo aproximado de una hora a la realización de pruebas parciales de la herramienta, de forma que se pueda ir probando la herramienta durante su implementación en busca de errores y evitar su propagación.

7.2. Presupuesto

En este apartado serán calculados los gastos previstos para la realización del Trabajo Fin de Grado. Serán detallados tanto recursos materiales como humanos, además de los costes indirectos producidos y los gastos en viajes y dietas generados.

7.2.1. Recursos materiales

En este punto se expondrán los costos asociados a los equipos utilizados y software necesario para el desarrollo e implantación de la herramienta creada. Únicamente serán reflejados los materiales que hayan supuesto un costo.

Recurso	Cantidad	Coste (€)
Ordenador	1	740,00
Pen Drive	1	5,50
Microsoft Office 2010	1	119,00
Alquiler Servidor Web	1 x 12 meses	2090,52
Total		2955,02

Tabla 67. Resumen de costes en recursos materiales

En el cálculo del presupuesto aportado para el Servidor Web, donde se alojará la lógica de negocio y la base de datos, se ha tenido en cuenta el periodo mínimo de alquiler necesario para el desarrollo del proyecto.

7.2.2. Recursos humanos

Una vez conocemos la *Planificación* del proyecto, donde está previsto que el trabajo sea desarrollado a lo largo de setenta (70) días, sin tener en cuenta fines de semana ni festivos, con una media de seis (6) horas de dedicación cada uno de esos días, podemos conocer el número aproximado de horas empleadas al mismo.

Anualmente la Fundación de la Universidad Carlos III realiza un estudio sobre la inserción en el mundo laboral de los recién titulados por la Universidad Carlos III [28]. Centrándonos en el último de estos estudios, elaborado en el año 2012 sobre la promoción de 2010, podemos observar que el salario medio oscila en torno a los 1400 € netos/mes.

Conociendo que el número de días laborables anualmente es de doscientos cincuenta y dos (252), los cuales son repartidos en doce (12) meses, por lo que cada mes tendría una media de veintiún (21) días laborables, y suponiendo una jornada de trabajo media de ocho (8) horas, podemos deducir una media de ciento sesenta y ocho (168) horas trabajadas mensualmente.

De las suposiciones anteriores podemos calcular que un recién titulado tiene un salario por hora aproximado de 8,33 €. En base a este cálculo desglosaremos el gasto en recursos humanos previsto para el proyecto.

Fase	Horas	€/hora	Coste (€)
Planteamiento	32	8,33	266,67
Estado del Arte	60	8,33	500,00
Análisis del Sistema	52	8,33	433,33
Diseño del Sistema	28	8,33	233,33
Implementación	76	8,33	633,33
Pruebas	16	8,33	133,33
Documentación	156	8,33	1300,00
Total	420	8,33	3500,00

Tabla 68. Resumen de costes en recursos humanos

7.2.3. Viajes y dietas

Teniendo en cuenta las distintas reuniones programadas con el tutor del Trabajo Fin Grado, cuyo despacho se encuentra en Leganés (Madrid) y la distancia existente con el inmueble donde se desarrolla el mismo, es imprescindible incluir los gastos de transporte para el desplazamiento a dichas reuniones.

Reuniones	Distancia (km)	Viaje (€)	Coste total (€)
7	50	12,00	84,00
Total			84,00

Tabla 69. Resumen de costes de viajes y dietas

7.2.4. Gastos indirectos

Debemos tener en cuenta los gastos derivados del proceso de producción de la herramienta que no pueden aplicarse como generados directamente por el proyecto, pero que de no existir no sería posible la realización del mismo.

Concepto	Coste (€/mes)	Periodo de uso (mes)	Coste imputable (€)
Conexión Internet	39,90	4	159,60
Consumo energético	45,00	4	180,00
Total			339,60

Tabla 70. Resumen de costes indirectos

7.2.5. Presupuesto total

Unificamos las tablas desglosadas anteriormente en una tabla resumen para obtener el coste al que asciende el presupuesto del Trabajo Fin de Grado, al cual habrá que aplicar el 21% de I.V.A. para la obtención del presupuesto total.

Tipo Gasto	Coste (€)
Recursos materiales	2955,02
Recursos humanos	3500,00
Viajes y dietas	84,00
Gastos indirectos	339,60
Subtotal	6878,62
I.V.A. (21%)	1444,51
Total	8323,13

Tabla 71. Resumen del Presupuesto Total

8. CONCLUSIONES Y TRABAJO FUTURO

8.1. Conclusiones

Atrás han quedado los días en los que navegábamos en Internet, leíamos páginas web y enviábamos los enlaces de los sitios interesantes que habíamos encontrado por correo electrónico a nuestros amigos.

En la actualidad, con el surgimiento de las redes sociales, hacemos clic en los botones encontrados en las páginas web para indicar qué “me gusta”, qué deseamos “compartir” o “recomendar”, publicamos un “tweet” mostrando enlaces de páginas web que nos han llamado la atención a nuestros seguidores o hacemos “retweet” de otras publicaciones que nos han causado alguna curiosidad.

De esta forma los usuarios de las redes sociales dejan muestras de páginas web donde han encontrado contenidos interesantes. Estas señales podrán ser utilizadas por sus “círculos”, “amigos” o “seguidores” para acceder a sitios web que les pueden resultar atractivos.

Sin embargo, estas “feromonas” depositadas por los usuarios de las redes sociales únicamente podrán ser aprovechadas por el resto de usuarios con los que están relacionados. Y en otras muchas ocasiones, la huella dejada no causa ninguna ventaja para los muchos usuarios que tienen que ver estos enlaces sin que resulten de su interés.

Nuestro prototipo presenta una idea de búsqueda web mejorada aprendiendo de las hormigas. Los usuarios dejarán “feromonas” en su camino hacia una “fuente de alimentación” con información o contenido de su interés.

La ventaja consiste en que el usuario aún es el que inicia la consulta al introducir unas palabras clave en el motor de búsqueda, es decir, la información proporcionada será relevante en su búsqueda. Asimismo no será únicamente un rastro aprovechable por los usuarios relacionados, las sugerencias y experiencias estarán también disponibles para toda la comunidad de usuarios que naveguen en la web.

En el aspecto personal debo considerar el proyecto realizado, un trabajo muy productivo para mi formación en lo que respecta a la captación de nuevos conocimientos y a la aplicación de una gran variedad de aptitudes adquiridas en la Universidad a lo largo de los últimos años.

8.2. Trabajo futuro

Por último citar los posibles trabajos futuros para la mejora y/o ampliación de la herramienta:

- Tenemos interés en que nuestra herramienta pueda ser aprovechada por el mayor número de usuarios posibles. En el mercado existe una amplia variedad de navegadores y en la actualidad la mayoría permite complementar su funcionalidad con extensiones. Por ésta razón en lo que respecta a la plataforma, se pretende que la herramienta pueda estar presente también en otros navegadores.
- Por otra parte, el prototipo actual trabaja con búsquedas exactas, es decir, los resultados mostrados están basados en consultas con las palabras clave idénticas. Se pretende que en un futuro la herramienta tenga la capacidad de trabajar con búsquedas similares.
- De igual forma se trabajará para añadir al sistema la capacidad de que los usuarios puedan elegir los temas de su interés formando una especie de “colonia” de usuarios por cada uno de esos temas. Así usuarios con mayor interés en un tema, colaborarán conjuntamente para encontrar información más específica.
- Un próximo desarrollo de la herramienta debe ser la protección frente a posibles “ataques” realizados por desarrolladores de sitios web que al igual que sucede con *Search Engine Optimization (SEO)* en PageRank traten de mejorar sus resultados de relevancia. Una alteración de los resultados puede producirse cuando a un desarrollador le interesa subir puestos en el ranking de resultados y para ello realiza constantes acciones sobre su página, como puede ser marcar repetidamente su página como favorito, provocando que los resultados no tengan un carácter social al no estar basados en consultas verdaderas.
- Actualmente, todas las acciones contempladas aportan el mismo valor de feromona a la solución. Se pretende realizar un estudio para mejorar el modelo de actualización de la feromona, donde se tenga en cuenta la frecuencia de las acciones realizadas y otorgar más peso a unas u otras dependiendo de ésta estadística.

9. BIBLIOGRAFÍA

1. SERGEY, Brin y LAWRENCE, Page. *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, 1998. [<http://infolab.stanford.edu/~backrub/google.html>]
2. BOSCH, Sara. *El algoritmo del PageRank de Google y su funcionamiento*, 2008. [<http://www.posicionapro.com/blog/wp-content/uploads/2008/11/formula-pagerank-google.pdf>]
3. SEOmoz. *Google Algorithm Change History*. [<http://www.seomoz.org/google-algorithm-change>]
4. KLEINBERG, J. M. *Authoritative sources in hyperlinked environment*. En: ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS, 1998. [<http://simon.cs.cornell.edu/home/kleinber/auth.ps>]
5. SCHWARTZ, Barry. *Goodbye Teoma Algorithm, Hello Edison, Says Ask.com*, 2007. [<http://searchengineland.com/goodbye-teoma-algorithm-hello-edison-says-askcom-10962>]
6. WANG, Z. *Improved Link-Based Algorithms for Ranking Web Pages*. En Proc. of The Fifth International Conference on WebAge Information Management, LNCS 3129, pp. 291-302, 2004.
7. *Guía para principiantes sobre optimización para motores de búsqueda*, 2011. [https://docs.google.com/viewer?url=http://www.google.es/webmasters/docs/guia_optimizacion_motores_busqueda.pdf&pli=1]
8. *An update to Google Social Search*, 2011. [<http://googleblog.blogspot.com.es/2011/02/update-to-google-social-search.html>]
9. ORIHUELA, José Luis. *De buscar información a encontrar personas*, 2011. [<http://www.abc.es/blogs/jose-luis-orihuela/public/post/de-buscar-informacion-a-encontrar-personas-9907.asp>]
10. ORIHUELA, José Luis. *La hora de los algoritmos sociales*, 2011. [<http://www.abc.es/blogs/jose-luis-orihuela/public/post/la-hora-de-los-algoritmos-sociales-10356.asp>]
11. GALÁN, Sergio M. *Filtrado Colaborativo y Sistemas de Recomendación*, 2007. [<http://www.it.uc3m.es/jvillena/irc/practicass/06-07/31.pdf>]

12. KINCAID, Jason. *EdgeRank: The Secret Sauce That Makes Facebook's News Feed Tick*, 2010. [<http://techcrunch.com/2010/04/22/facebook-edgerank/>]
13. KENNEDY, James y EBERHART, Russell. *Particle Swarm Optimization*, 1995. [http://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6%201995%20particle%20swarming.pdf]
14. KLAUS JAFEE C. *El Mundo de las Hormigas*. EQUINOCCIO, Ediciones de la Universidad Simón Bolívar, ISBN: 980-237-058-4, 1993.
15. DENEUBOURG, J.L., ARON, S., GOSS, S., PASTEELS, J.M. *The Self-Organizing Exploratory Pattern of the Argentine Ant*. Journal of Insect Behavior, 3, pp. 159 – 168, 1990.
16. DORIGO, M., MANIEZZO, V. y COLORNI, A. *The Ant System: Optimization by a Colony of Cooperating Agents*, IEEE Transactions on Systems, Man and Cybernetics, 26, n.1, pp. 29-42, 1996.
17. PIROLI, P., CARD, S. *Information Foraging*, Psychological Review, vol. 106(4), pp. 643-675, 1999.
18. MALIZIA, A., OLSEN, K. *Emerging social search paradigms: can we learn from ants?* Proc. of the Italian Workshop on Artificial Life and Evolutionary Computation 1- 4, 2012. [http://wivace2012.ce.unipr.it/Papers/malizia_al.pdf]
19. OLSEN, K., MALIZIA, A. *Following virtual trails*. Potentials, IEEE 29(1). 24-28, 2010.
20. Chrome Web Store. *¿Qué son las aplicaciones, las extensiones y los temas?* [http://www.google.com/intl/es_ALL/landing/chrome/webstore/create/whats-in-store.html]
21. ESA Software Engineering Standards, PSS-05-0 [<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS050.pdf>]
22. Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. [http://noticias.juridicas.com/base_datos/Admin/lo15-1999.html]
23. *Getting Started: Building a Chrome Extension*. [<https://developer.chrome.com/extensions/getstarted.html>]
24. *Introducción a JSON*. [<http://www.json.org/json-es.html>]

25. AEPD. *Declaración sobre buscadores en Internet*, 2007.
[https://www.agpd.es/portalwebAGPD/canaldocumentacion/recomendaciones/common/pdfs/declaracion_aepd_buscadores.pdf]
26. IEEE Standard for Software and System Test Documentation. *IEEE Std 829-2008*, 2008.
27. Universidad Carlos III de Madrid. *Revisión de Métrica Versión 3*.
28. Fundación Universidad Carlos III. *XVI Estudio de Inserción Profesional de los Titulados de la Universidad Carlos III de Madrid, Promoción 2010*.
[http://e-archivo.uc3m.es/bitstream/10016/14286/1/resumen_fundacion_EIPXVI.pdf]

ANEXO I: ACRÓNIMOS Y ABREVIATURAS

- **ACO:** *Ant Colony Optimization (Optimización basada en Colonias de Hormigas)*
- **AEPD:** *Agencia Española de Protección de Datos*
- **AJAX:** *Asynchronous JavaScript And XML (JavaScript Asíncrono y XML)*
- **API:** *Application Programming Interface (Interfaz de Programación de Aplicaciones)*
- **AS:** *Ant System (Sistema de Hormigas)*
- **CSS:** *Cascading Style Sheets (Hojas de Estilo en Cascada)*
- **DOM:** *Document Object Model (Modelo de Objetos del Documento)*
- **DoS:** *Denial of Service (Denegación de Servicio)*
- **HITS:** *Hyperlink Induced Topic Search*
- **HTML:** *HyperText Markup Language (Lenguaje de Marcado de Hipertexto)*
- **HTML5:** *HyperText Markup Language v5 (Lenguaje de Marcado de Hipertexto v5)*
- **HTTP:** *HyperText Transfer Protocol (Protocolo de Transferencia de Hipertexto)*
- **Internet:** *International Network*
- **IVA:** *Impuesto sobre el Valor Añadido*
- **JSON:** *JavaScript Object Notation (Notación de Objetos JavaScript)*
- **LOPD:** *Ley Orgánica de Protección de Datos*
- **LSSI:** *Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico*
- **PSO:** *Particle Swarm Optimization (Optimización de Enjambre de Partículas)*
- **SALSA:** *Stochastic Approach for Link Structure Analysis*
- **SEO:** *Search Engine Optimization (Posicionamiento en Motores de Búsqueda)*
- **SQL:** *Structured Query Language (Lenguaje de Consulta Estructurado)*
- **TFG:** *Trabajo Fin de Grado*
- **URL:** *Uniform Resource Locator (Localizador de Recursos Uniforme)*
- **WWW:** *World Wide Web (Red Informática Mundial)*

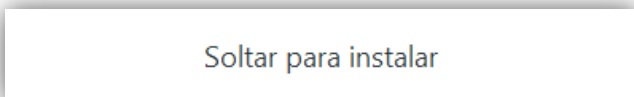
ANEXO II: MANUAL DE USUARIO

Virtual Ants, su herramienta para encontrar los resultados que busca en Google. Le ofrece las páginas web más relevantes para otros usuarios que realizaron su misma consulta. Con sus acciones en busca de un contenido de calidad tras realizar una búsqueda le permite aportar a la colonia de usuarios los sitios web que le han resultado más interesantes.

Instalación

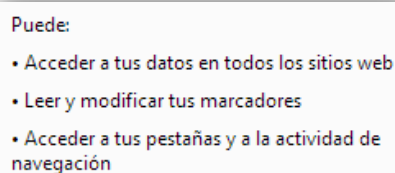
Para instalar **Virtual Ants** en su navegador Google Chrome necesita disponer del fichero “VirtualAnts.crx” y seguir los siguientes pasos:

1. Abra su navegador Google Chrome y sitúese en <chrome://extensions/>.
2. Arrastre el fichero “VirtualAnts.crx” sobre el sitio mencionado hasta que aparezca el siguiente mensaje:



Soltar para instalar

3. Una vez haya soltado el archivo se le pedirá confirmación con la pregunta “¿Quieres instalar “Virtual Ants”?” y se le informará que los siguientes permisos son necesarios:




Puede:

- Acceder a tus datos en todos los sitios web
- Leer y modificar tus marcadores
- Acceder a tus pestañas y a la actividad de navegación

4. Deberá hacer clic sobre el botón  para completar la instalación.

Empezar a utilizar la extensión

En primer lugar será necesario iniciar sesión en la aplicación. Para ello debes hacer clic sobre el icono de la extensión  que se encuentra en la barra de herramientas del navegador.



Si aún no estás registrado en la aplicación y por lo tanto no puedes realizar el inicio de sesión debes pulsar sobre “**Regístrate**” y se abrirá la siguiente ventana donde se le requerirá únicamente un correo electrónico válido y una contraseña para efectuar el registro:



Email

Contraseña

Confirmar contraseña

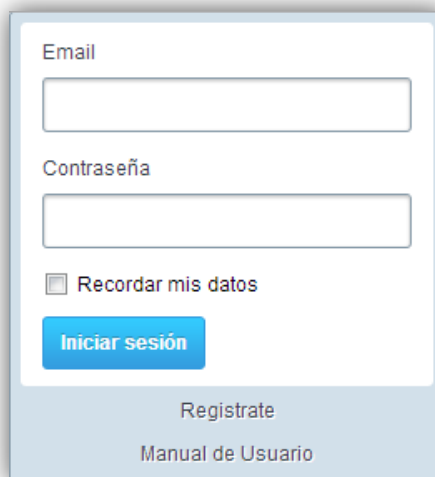
☐ Aceptas la [Política de Privacidad](#)

Registrarse

[Página de Inicio](#)

Para poder completar el registro deberás aceptar la Política de Privacidad de **Virtual Ants**.

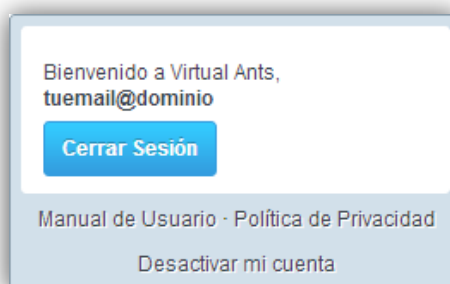
Una vez eres usuario registrado, pulsado el icono de la barra de herramientas se abrirá la siguiente ventana emergente donde introducir el correo electrónico y la contraseña para realizar el inicio de sesión:



The login form is a light blue box with a white background. It contains two input fields: 'Email' and 'Contraseña'. Below the 'Contraseña' field is a checkbox labeled 'Recordar mis datos'. A blue button labeled 'Iniciar sesión' is positioned below the checkbox. At the bottom of the form, there are two links: 'Registrate' and 'Manual de Usuario'.

En caso de querer que la herramienta te recuerde en el equipo de forma que no sea necesario introducir los datos para el inicio de sesión cada vez que pretendas hacer uso de la aplicación, tendrás la posibilidad de marcar “Recordar mis datos”.

Cuando el inicio de sesión se ha realizado correctamente se mostrará la siguiente página de bienvenida:

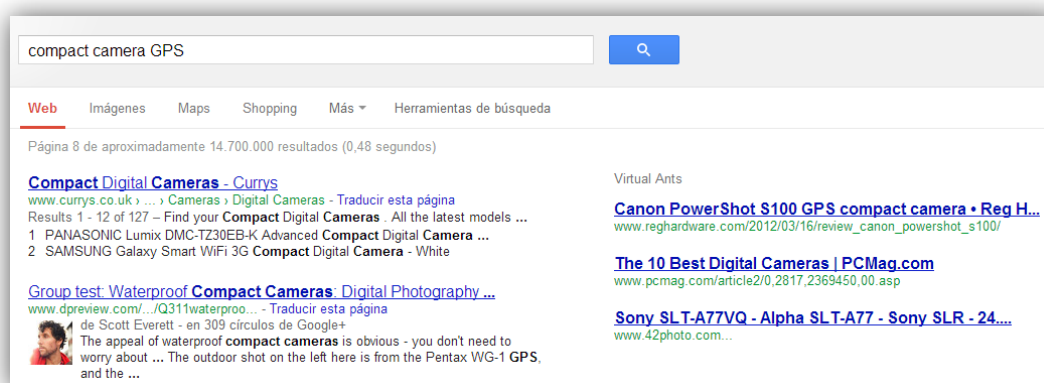


The welcome page is a light blue box with a white background. It displays the text 'Bienvenido a Virtual Ants, tuemail@dominio'. Below this text is a blue button labeled 'Cerrar Sesión'. At the bottom of the page, there are two links: 'Manual de Usuario · Política de Privacidad' and 'Desactivar mi cuenta'.

Realizar una búsqueda en Google

Para ver los resultados ofrecidos por la herramienta únicamente necesitarás realizar una búsqueda en Google de igual forma que cuando la herramienta no es utilizada. En la página de resultados seguirás teniendo los enlaces ofrecidos por Google, pero adicionalmente observarás una columna alternativa con el ranking de resultados ofrecidos por **Virtual Ants**.

A continuación mostramos una imagen de una búsqueda realizada en Google cuando la herramienta está siendo utilizada:



Acciones sobre las páginas

Para colaborar con el resto de “hormigas virtuales” en la creación de unos resultados de búsqueda sociales y dinámicos, los usuarios con la extensión instalada y sesión iniciada iréis dejando un rastro en la ruta seguida hasta encontrar una página web con un contenido interesante sobre la consulta realizada previamente.

La herramienta detectará que un sitio web ofrece información interesante cuando los usuarios impriman la página, la añadan a *favoritos* o la compartan en redes sociales.

ANEXO III: POLÍTICA DE PRIVACIDAD

POLÍTICA DE PRIVACIDAD Y PROTECCIÓN DE DATOS PERSONALES

Previamente a la creación de una cuenta y pasar a ser un usuario de la herramienta privada **Virtual Ants**, debes leer esta Política de Privacidad y Protección de Datos Personales (en adelante, "Política de Privacidad") y aceptarla marcando la casilla establecida en el formulario de registro o mediante el procedimiento equivalente que se establezca, en su caso.

Nuestra Política de Privacidad afecta tanto a la información personal que nos proporcionas al crearte un Perfil como a todos los datos que nos facilites al acceder y hacer uso de cualquiera de los servicios disponibles en la herramienta **Virtual Ants** durante el tiempo que dispongas de una cuenta.

Los datos que aportas a **Virtual Ants** son incluidos en nuestros ficheros registrados ante la Agencia Española de Protección de Datos (en adelante, "AEPD"). La AEPD se encarga de velar por el cumplimiento de las leyes españolas sobre privacidad y protección de datos y de garantizar la seguridad y privacidad de tus datos personales.

Virtual Ants recoge y trata tus datos para identificarte como usuario de la herramienta y para darte acceso a los servicios y funcionalidades que ponemos a tu disposición.

Como usuario tienes derecho a acceder a tus datos personales para saber cómo los tratamos y con qué finalidad. Además, podrás rectificar cualquier error en tus datos si ves que no son correctos o exactos e incluso podrás pedirnos que los cancelemos si quieres dejar de ser usuario de **Virtual Ants**.

© Virtual Ants, 2013.